UNIVERSITÉ FRANÇOIS-RABELAIS
TOURS

École Doctorale : Santé, Sciences et Technologies (SST)

Laboratoire d'Informatique (LI)

THÈSE présentée par :

Sonia COLAS

soutenue le : 17 novembre 2008

pour obtenir le grade de **Docteur de l'Université François-Rabelais**

Discipline : Informatique

OUTILS D'AMÉLIORATION DE L'ACCESSIBILITÉ DU WEB POUR LES PERSONNES VISUELLEMENT HANDICAPÉES

THÈSE dirigée par :

M. Mohamed SLIMANE Professeur Université de Tours

RAPPORTEURS :

M.	Jaime	LOPEZ KRAHE	Professeur	Université Paris 8
M.	El-Ghazali	TALBI	Professeur	Université de Lille 1

JURY :

M.	Jaime	LOPEZ KRAHE	Professeur	Université Paris 8
M.	Nicolas	MONMARCHÉ	Maître de Conférences	Université de Tours
M.	Patrick	SIARRY	Professeur	Université Paris 12
M.	Mohamed	SLIMANE	Professeur	Université de Tours
M.	El-Ghazali	TALBI	Professeur	Université de Lille 1
Mme	Nadine	VIGOUROUX	Chargée de Recherches CNRS	Toulouse

Remerciements

Ce travail a été effectué au Laboratoire d'Informatique (LI) de Tours, plus précisément dans l'équipe HaNT que je remercie grandement pour son accueil durant ces années et aussi pour avoir créé une ambiance d'entraide très agréable. Dans cette équipe, ma gratitude va plus particulièrement à Mohamed Slimane et à Nicolas Monmarché pour avoir encadré ce travail de thèse avec beaucoup d'enthousiasme et de disponibilité. Merci à vous pour vos précieux conseils et pour la confiance que vous m'avez accordée au cours de cette tranche de vie.

Je tiens également à remercier les étudiants de Polytech'Tours pour les nombreux projets qu'ils ont menés, notamment Marc Galvan, Emmanuelle Launay, Benoit Thomas, Antoine Blais et Jérôme Bulucua pour leurs de projets de fin d'études.

Je ne saurai oublier mes collègues de la faculté de Droit, avec qui j'ai travaillé à mi-temps au service des emplois du temps en parallèle à ma thèse. Je remercie plus spécialement Florence Viollette et Annette Opitz pour l'atmosphère chaleureuse et sympathique dont elles ont su m'entourer et pour leur compréhension concernant mes obligations d'enseignement ou de conférences ; ainsi que Carole Vourc'h qui occupait le bureau avec moi, et Christophe Boussy avec qui je partageais les nombreuses pauses café.

Je remercie ensuite M. Jaime Lopez Krahe, Professeur à l'Université de Paris 8, responsable de l'équipe THIM (Technologies, Handicaps, Interfaces, Multimodalités), ainsi que M. El-Ghazali Talbi, Professeur à l'Université de Lille 1, responsable de l'équipe OPAC (Optimisation PArallèle et Coopérative) au LIFL (Laboratoire d'Informatique Fondamentale de Lille), qui ont accepté de juger ce travail et d'en être les rapporteurs. Merci également à Mme Nadine Vigouroux, et à M. Patrick Siarry pour avoir bien voulu examiner mon mémoire et faire partie de mon jury de thèse.

Enfin, pour leur soutien sans faille et permanent, je tiens à remercier de tout cœur mes parents, ma sœur et Julien.

Résumé

Avec l'essor d'Internet, de nombreux services ont été mis en place, tels des services commerciaux et des services administratifs, pour obtenir plus rapidement et plus facilement des documents, des renseignements, des produits... Pour profiter de ces services et « surfer » sur la toile comme quiconque, les personnes handicapées ont recours à des aides techniques, par exemple le clavier braille et la synthèse vocale pour les non-voyants. Des normes ont été établies pour assurer la compatibilité des sites web avec les aides techniques, et des lois ont été instaurées pour en imposer le respect (notamment la loi *pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées*, votée en 2005 en France). Néanmoins encore trop peu de sites web respectent ces normes et sont de ce fait inaccessibles aux personnes handicapées.

Lors de ce travail, nous avons cherché à améliorer l'accessibilité du web aux personnes handicapées, d'une part en contribuant à compenser les problèmes liés directement à la non-conformité des pages web et d'autre part en augmentant les performances des interfaces Homme-Machine. Nous abordons dans cette thèse deux notions, à savoir l'adaptation « du » web (ou adaptation du contenu) ainsi que l'adaptation « au » web (ou adaptation de l'utilisateur), qui sont indispensables pour que les personnes handicapées puissent pleinement disposer des services offert par Internet.

D'une part, l'adaptation du web relève de la responsabilité des créateurs de contenu pour le web. L'accessibilité de l'information d'une page dépend bien évidemment des possibilités offertes par le lecteur d'écran, mais aussi du contenu et de la disposition de la page. Par conséquent de nombreux efforts sont à la charge du webmestre qui doit alors fournir un site web accessible, c'est-à-dire respectant les normes établies. Afin de faciliter le travail des webmestres, nous avons tout d'abord voulu participer à la diffusion d'informations concernant l'accessibilité du web pour que les règles soient connues par un public toujours plus large. Ensuite, nous avons choisi de créer des outils (guide, validateur, correcteur) favorisant la démarche vers l'accessibilité des sites web des petites collectivités, pour lesquelles l'intervention d'un informaticien qualifié dans ce domaine est souvent exclue pour des raisons financières.

D'autre part, l'adaptation de l'utilisateur au web consiste à faciliter l'accès au web pour les personnes handicapées, ce qui s'avère être un point crucial en attendant la conformité (utopique) de l'ensemble des sites web. A cet effet nous avons mis en place des systèmes visant à améliorer automatiquement le niveau d'accessibilité des sites web, que ce soit au regard des normes, en proposant des moyens de navigation alternatifs à ceux proposés nativement sur le site (lorsqu'ils existent), ou encore en accélérant la vitesse d'accès aux informations, laquelle pouvant être réduite de par l'utilisation de certaines aides techniques.

Mots clés : informatique pour la compensation des déficits d'autonomie, accessibilité du web, handicap, malvoyant, non-voyant, fourmis artificielles

Abstract

With the rapid growth of the web, many services have been proposed to users. Among them there are commercial services (online purchases, plane or train ticket reservation...) and administrative services. Disabled persons use assistive technologies (like braille keyboard or voice synthesis for blind people) to take advantage of numerous services offered by Internet and to navigate on the World Wide Web as whoever. Standards were established to insure compatibility of Web sites with assistive technologies, and laws were established to impose respect of these standards (notably the law titled "pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées", passed in 2005 in France). However few Web sites respect these standards and as a result a lot of Web sites are inaccessible for disabled persons.

During this work, we tried to improve web accessibility to disabled people, by contributing to compensate for the problem of accessibility Web pages and to enhance quality of user interfaces. We approach two notions in this thesis, namely the web adaptation (or contents adaptation) as well as the adaptation to web (or user adaptation).

On the one hand, the web adaptation is from the responsibility of web contents creators, which have to supply an accessible Web site (respecting the established standards). To facilitate the webmasters work, we wanted first participate in information circulation about web accessibility in order to many people know accessibility rules. After we contributed to facilitate web adaptation by proposing for example tools to help webmasters to return their accessible site towards the standards (validator, corrective tool) and to support them during the carrying out global structure of Web site .

On the other hand, the user adaptation to web consists in facilitating web access for the disabled people, what turns out to be a crucial point by waiting for the (utopian) compliance of all Web sites. For that we made tools to improve automatically the level of accessibility of Web sites, whether it is towards the standards, by proposing alternative means of navigation to those proposed on the site (when they exist), or still by accelerating the speed of access to information.

Keywords : computer science to compensate for autonomy deficiency, web accessibility, people with disabilities, visually handicapped people, blind people, artificial ant

Table des matières

Liste des figures **xvii**

Introduction **1**

I Etats de l'art 7

1 L'accessibilité du web **9**

1.1 Introduction . . . 9

1.2 Présentation générale du handicap . . . 11

1.2.1 Définition et chiffres sur le handicap . . . 11

Classification des handicaps . . . 11

Les chiffres du handicap en France . . . 13

Classification des handicaps visuels . . . 14

Les chiffres du handicap visuel en France . . . 16

1.2.2 Les handicapés visuels et l'accès à la culture écrite . . . 16

Les premiers systèmes d'écriture en relief . . . 16

La méthode actuelle : le braille . . . 19

1.2.3 Les handicapés visuels et l'informatique . . . 23

Les aides techniques . . . 23

Les handicapés visuels et Internet . . . 25

1.3 Cadre légal et organisationnel de l'accessibilité du web . . . 29

1.3.1 Les organismes et associations travaillant sur l'accessibilité . . . 29

Le W3C et la WAI . . . 29

Les projets dans les pays européens . . . 32

La situation en Europe . . . 33

1.3.2 La réglementation concernant l'accessibilité . . . 34
Aux Etats-Unis . . . 34
Dans certains pays européens . . . 35
Pour toute l'Europe . . . 36
1.4 En pratique : créer ou rendre un site web accessible . . . 38
1.4.1 Accessibilité et utilisabilité . . . 38
Relation entre accessibilité et utilisabilité . . . 38
Règles générales d'utilisabilité des sites web . . . 38
1.4.2 Les guides : synthèse sur l'accessibilité des sites web . . . 40
1.4.3 Les validateurs : vérification automatique . . . 40
Tester le respect des normes HTML . . . 40
Tester le respect des normes CSS . . . 42
Tester l'accessibilité . . . 42
1.4.4 Les outils facilitant la vérification manuelle . . . 44
Observation avec différents navigateurs . . . 45
Observation avec différentes résolutions . . . 46
Test avec des outils d'aide à l'accessibilité . . . 46
1.5 Conclusion . . . 50

2 Se repérer/se diriger dans un site et s'exprimer à l'écrit 53
2.1 Introduction . . . 53
2.2 Faciliter l'orientation dans un site grâce au plan du site . . . 55
2.2.1 Les pages web dédiées à l'affichage d'un plan du site . . . 55
2.2.2 Les générateurs de plans de sites . . . 60
2.3 Faciliter l'écriture grâce aux aides techniques de substitution du clavier d'ordinateur standard . . . 69
2.3.1 Les claviers matériels modifiés . . . 69
Les claviers ergonomiques . . . 70
Les claviers alternatifs . . . 72
2.3.2 Les claviers virtuels . . . 75
La sélection des touches . . . 76
Les systèmes de prédiction . . . 78
La réorganisation des touches . . . 80
2.4 Conclusion . . . 83

3 Les fourmis artificielles 85

3.1 Introduction 85

3.2 Les fourmis réelles 86

3.2.1 Généralités 86

3.2.2 Les différentes tâches réalisées par les fourmis 86

3.2.3 Organisation des fourmis en castes 86

3.2.4 La communication 87

3.3 Naissance des fourmis artificielles 89

3.3.1 Les fourmis et le problème du voyageur de commerce 89

3.3.2 De Ant System à Ant Colony System 89

3.3.3 Heuristique ACO 91

3.4 Résolution de problèmes d'optimisation combinatoire par les fourmis artificielles 92

3.4.1 Le « Vehicle Routing Problem » (VRP) 92

3.4.2 Le « Minimum Spanning Tree » (MST) 93

Le « Generalized Minimum Spanning Tree » (GMST) 93

Le « Degree-Constrained Minimum Spanning Tree » (d-MST) 94

Le « Capacitated Minimum Spanning Tree » (CMST) 95

3.4.3 Le « Keyboard Arrangement Problem » (KAP) 95

3.5 Conclusion 97

II Nos contributions à l'accessibilité du web 99

4 Outils d'aide à l'accessibilité 101

4.1 Introduction 101

4.2 Site web d'aide à la migration 103

4.2.1 Les ressources nécessaires pour bien démarrer 103

4.2.2 Mode d'emploi pour la création ou la migration de site accessible . . 105

4.2.3 Divers outils pratiques 106

4.2.4 Application 109

4.3 Traducteur en braille abrégé 111

4.3.1 Méthode de transcription 111

4.3.2 Résultats expérimentaux 112

4.4 Optimisation d'un clavier virtuel par des fourmis artificielles 115

4.4.1 Modélisation et algorithme . . . 115
4.4.2 Construction de la solution . . . 116
4.4.3 Evaluation de la solution . . . 117
4.4.4 Mise à jour des phéromones . . . 118
4.4.5 Résultats expérimentaux . . . 118
4.5 Conclusion . . . 124

5 Outils de synthèse **125**
5.1 Introduction . . . 125
5.2 Validateur multi-recommandations . . . 127
5.2.1 Présentation du validateur . . . 127
5.2.2 Fonctionnement et structure du validateur . . . 128
Choix des validateurs . . . 129
La base de données . . . 129
5.2.3 Les normes supplémentaires . . . 130
5.2.4 Affichage des résultats . . . 132
Choix des erreurs à afficher . . . 132
Liste des erreurs . . . 133
5.3 Génération de plans de sites web par des fourmis artificielles . . . 135
5.3.1 Modèle biologique et informatique des fourmis artificielles . . . 136
5.3.2 Algorithme principal du générateur de plan . . . 138
5.3.3 Rôle des fourmis artificielles . . . 139
Génération des h circuits . . . 139
Les phéromones . . . 141
Le test d'arrêt . . . 142
5.3.4 Application et résultats expérimentaux . . . 142
5.4 Conclusion . . . 145

6 Outils de transformation de pages **147**
6.1 Introduction . . . 147
6.2 Correcteur de pages web . . . 148
6.2.1 Le moteur . . . 148
6.2.2 Les plugins . . . 150
6.2.3 Choix des plugins implémentés . . . 151

6.2.4 Exemples des plugins de la directive 5 des WCAG 152
Etude de la directive 5 152
Correction des erreurs pour chacune des spécifications 153
Traitement additionnel spécifique à la directive 154
Interfaces de plugins de la directive 5 154
6.2.5 Exemple d'application et résultats 156
Exemple d'application 156
Résultats obtenus sur les sites web des communes d'Indre-et-Loire . . 157
6.3 Réorganisation de pages web 161
6.3.1 Construction de l'arbre représentatif de la page 162
6.3.2 Détection des parties redondantes sur les pages 163
6.3.3 Détection des menus 164
6.3.4 Application et résultats expérimentaux 165
Valeur de θ 165
Exemple d'application 167
6.4 Conclusion 169

Conclusion **171**

A Algorithmes principaux utilisés lors de la transcription en braille abrégé 179

B Evaluation de l'accessibilité des sites web publics d'Indre-et-Loire 183

C Evaluation de l'accessibilité des sites web publics d'Indre-et-Loire après utilisation du correcteur automatique 191

Références bibliographiques 197

Références Internet 209

Liste des figures

1 Les principales contributions. 2
2 Regroupement des outils selon l'utilisateur. 5
3 Regroupement des outils selon leur rôle. 5
4 Organisation du document en deux parties et relations entre ces dernières. . 6

1.1 Séquence de WOOD. 11
1.2 Séquence de la CIF. 12
1.3 Alphabet de François Lana Terzi (1670). 17
1.4 Système phonétique de Charles Barbier (1821). 17
1.5 Alphabet braille (1837). 18
1.6 Représentation en braille de la lettre A. 19
1.7 Le braille abrégé : abréviation d'une locution. 20
1.8 Le braille abrégé : symbole de « petit » et « petite ». 20
1.9 Le braille musical : représentation des 7 notes de la gamme. 22
1.10 Le braille musical : représentation des valeurs de la note Do. 22
1.11 Barre d'outils IE Accessible pour Internet Explorer. 27
1.12 Barre d'outils Accessibar pour Firefox. 27
1.13 Navigateur web Simply Web 2000. 28
1.14 Navigateur web BrailleSurf. 28
1.15 Logos WAI des niveaux de conformité. 31
1.16 Les 4 principaux guides francophones sur l'accessibilité du web. 41
1.17 Visualisation d'un site web par The Wave. 44
1.18 Le modèle des boites. 45

2.1 Plan d'un site sous la forme d'une table des matières. 57
2.2 Plan d'un site sous la forme d'un tableau. 58

2.3 Plan d'un autre site sous la forme d'un tableau. 58
2.4 Plan d'un site hybride. 59
2.5 Représentation d'un site avec WebTracer. 60
2.6 Représentation d'un site avec GEOmi. 61
2.7 Représentation d'un site avec GEOmi après rotation. 62
2.8 Barre de navigation Navibar. 63
2.9 Exemple de Google Sitemap au format xml. 64
2.10 Représentation d'un site avec SortSite. 65
2.11 Représentation de plans de site générés par PowerMapper. 66
2.12 Représentation de plans générés par Ixsite. 67
2.13 Organisation des touches du clavier Dvorak. 70
2.14 Clavier ergonomique fixe. 71
2.15 Claviers réglables. 71
2.16 Clavier sculptés. 72
2.17 Claviers grand format. 73
2.18 Clavier grand format avec diverses organisations de touches. 74
2.19 Guide-doigt. 74
2.20 Mini-claviers. 75
2.21 Clavier pour main droite. 75
2.22 Clavier virtuel de Microsoft. 77
2.23 Clavier CVK. 79
2.24 Clavier Keyglasses. 79
2.25 Clavier KNITS au format 5×6. 80
2.26 Clavier « KNITS » au format 4×10. 81
2.27 Clavier Métropolis. 81

3.1 Exploitation d'une source de nourriture par des fourmis. 88
3.2 Exemple d'ACM. 94

4.1 Page d'accueil de notre outil d'aide à la migration et à la création. 103
4.2 Les ressources. 104
4.3 Accéder au mode d'emploi personnalisé. 105
4.4 Mode d'emploi pour créer ou rendre un site web accessible. 106
4.5 Démarche pour rendre un site accessible. 107

4.6 Les divers outils. 107
4.7 Avancement dans les 11 fiches techniques. 108
4.8 Aide à la résolution d'une erreur HTML. 108
4.9 Résultat de l'analyse d'une page avec le validateur intégré à notre outil. . . . 109
4.10 Texte en braille intégral et abrégé. 113
4.11 Clavier de taille 12×3. 117
4.12 Entropie des phéromones. 119
4.13 Valeur des phéromones. 120
4.14 Evolution de la qualité du clavier. 121

5.1 Page d'accueil de notre validateur multi-recommandations. 127
5.2 Interface de lancement du validateur multi-recommandations. 128
5.3 Principe de fonctionnement du validateur. 129
5.4 MCD de la base de données du validateur. 130
5.5 MCD de la base de données du validateur (intégrant les normes Accessiweb et UWEM). 131
5.6 Interface permettant de choisir les erreurs à afficher. 132
5.7 Affichage des erreurs HTML. 133
5.8 Affichage des erreurs WGAG. 134
5.9 Affichage du plan pour le webmestre. 136
5.10 Diverses solutions de plans d'un site web. 137
5.11 Exemple de calcul de l'économie s_{ij}. 139
5.12 Génération de circuits par les fourmis artificielles. 140
5.13 Quantité de phéromones sur les liens partant de la racine. 143
5.14 Affichage du plan pour l'internaute. 144

6.1 Exemple de succession de traitements réalisable par notre correcteur. 148
6.2 Fonctionnement en mode console avec un proxy. 149
6.3 Interface du moteur. 149
6.4 Interface du moteur ayant chargé les plugins. 150
6.5 Répartition des erreurs et des warnings de sites web publics entre les directives WCAG. 151
6.6 Interface du plugin de correction de la spécification 5.3. 155
6.7 Interface du plugin de correction de la spécification 5.1. 155

6.8 La page web utilisée pour l'exemple d'application. 156
6.9 Interface du plugin permettant d'ajouter des textes alternatifs aux images (spécification 1.1). 157
6.10 Nombre d'erreurs par directive avant et après passage du correcteur. 158
6.11 Nombre d'erreurs par directive avant et après passage du correcteur (avec directive 3 et 11 tronquées). 158
6.12 Nombre de warnings par directive avant et après passage du correcteur. . . . 158
6.13 Nombre d'erreurs et de warnings concernant la directive 1 avant correction. . 159
6.14 Nombre d'erreurs et de warnings concernant la directive 1 après correction. . 159
6.15 Nombre d'erreurs et de warnings concernant la directive 5 avant correction. . 160
6.16 Nombre d'erreurs et de warnings concernant la directive 5 après correction. . 160
6.17 Exemple de site web avec ses différentes zones. 162
6.18 Schématisation de deux pages web d'un même site. 163
6.19 Arbres représentatifs des deux pages. 163
6.20 Arbres représentatifs des deux pages sans les zones redondantes. 164
6.21 Arbres représentatifs des deux pages sans les menus. 165
6.22 Menu du site du Ministère de l'Education Nationale. 166
6.23 Affichage du contenu propre à la page « La ville ». 167
6.24 Page « Loisirs » du site web de Chambray-les-Tours. 168
6.25 Affichage du contenu propre à la page « Loisirs ». 168
6.26 Les outils au service de l'internaute. 172
6.27 Les outils à destination du webmestre. 174
6.28 Représentation de l'ensemble des outils réalisés. 177

Introduction

Il y a encore quelque temps, les personnes malvoyantes et non-voyantes, en raison de leur handicap, étaient privées d'un accès à l'ordinateur. Grâce à l'émergence de nombreuses technologies d'interfaçage, la situation a fort heureusement évolué. Par l'intermédiaire de celles-ci, les personnes handicapées peuvent désormais utiliser en toute autonomie un ordinateur et devraient par extension pouvoir naviguer sur la toile comme quiconque. Néanmoins, en pratique les choses ne sont pas si simples.

Nous constatons effectivement que de nouvelles technologies se sont développées dans le but de rendre le web plus attractif visuellement (image, vidéo, flash...) mais entraînant en contrepartie la multiplication des sites web totalement inaccessibles à cause de leur incompatibilité avec les interfaces spécialisées dont les personnes handicapées ont besoin pour naviguer sur Internet.

Force est de remarquer qu'actuellement Internet est voué à jouer un rôle de plus en plus important dans nos vies quotidiennes. Ne pouvant exclure les personnes handicapées de ces avancées technologiques et de ce fait de la société, des normes ont été établies pour assurer la compatibilité des sites web avec les aides techniques, et des lois ont été instaurées pour en imposer le respect. Cependant ces normes, par leur complexité, nécessitent un recours à des informaticiens spécialisés et sensibilisés aux obstacles empêchant l'accès des personnes handicapées à Internet. Le surcoût que cela entraîne n'est pas toujours envisageable par les propriétaires de sites web modestes, ce qui, hélas, induit une mise en application totale des lois en faveur de l'accessibilité sur du long — voire très long — terme.

Vu le rôle majeur attribué à Internet dans la réalisation de tâches courantes, nous avons souhaité faciliter l'accès au web des personnes handicapées, que l'on se place du point de vue du webmestre (aides à la réalisation de sites web accessibles...) ou du point de vue de l'utilisateur (aides techniques, logiciels pour faciliter la navigation...).

Sujet de la thèse

Cette thèse de doctorat a été menée sous la direction du professeur Mohamed Slimane et le co-encadrement de Nicolas Monmarché, au sein de l'équipe Handicap et Nouvelles Technologies (HaNT) du Laboratoire d'Informatique (LI) de l'Université François Rabelais de Tours. Depuis sa création en 2003, l'équipe HaNT axe ses activités de recherche sur

les méthodes issues de la modélisation mathématique (chaînes de Markov cachées) et de l'intelligence artificielle (notamment les fourmis artificielles) appliquées à des problèmes de classification et d'optimisation, principalement dans le domaine des nouvelles technologies (web, réseaux...) et de l'aide aux personnes handicapées. Ainsi est venue l'idée de mettre à profit les compétences acquises dans les domaines du web, du handicap, et des fourmis artificielles, pour cette thèse qui s'intitule « Outils d'amélioration de l'accessibilité du web pour les personnes visuellement handicapées ». Elle s'est avérée être un sujet d'actualité puisque le *Projet de loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées* [7] a été adopté le 3 février 2005, et la loi en question a été votée le 12 février 2005. Un article concerne plus particulièrement l'accessibilité du web en imposant aux sites web publics de suivre les recommandations liées à l'accessibilité (section 1.3.2 page 34).

Lors de ce travail, nous avons cherché à améliorer l'accessibilité du web aux personnes handicapées, d'une part en contribuant à compenser les problèmes liés directement à la non-conformité des pages web et d'autre part en augmentant les performances des interfaces Homme-Machine. Les principales contributions de cette thèse sont illustrées sur la figure 1.

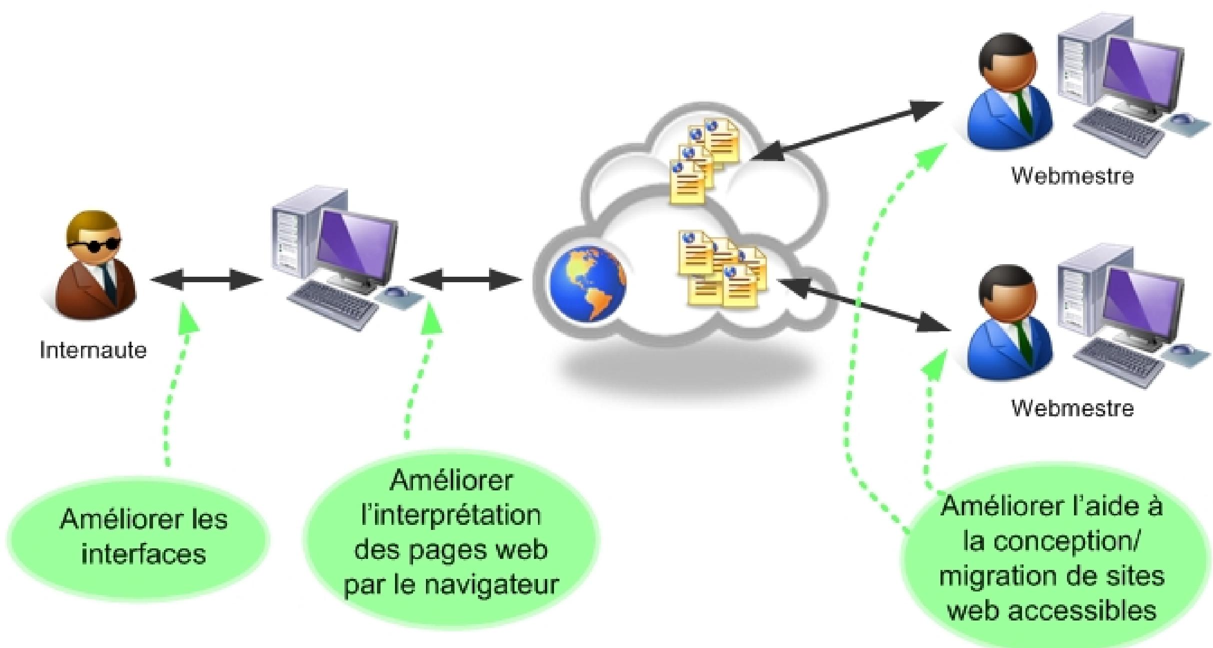


FIG. 1 – Les principales contributions.

Améliorer l'aide à la conception/migration de sites web accessibles

Une des raisons données par les webmestres pour expliquer le fait que peu de sites web soient accessibles, est le manque d'informations concernant l'accessibilité. Certaines règles d'accessibilité sont en effet simples à mettre en œuvre mais, n'étant pas connues des webmestres, ne sont pas respectées. Ainsi pour que tous soient informés sur les notions

élémentaires d'accessibilité, nous avons élaboré un site web d'aide à la création et à la migration de site internet accessible. Les webmestres peuvent y trouver toutes les bases utiles pour débuter. Certains webmestres se contenteront de ces informations pour commencer la modification de leurs sites web avec leurs logiciels habituels. D'autres, que nous appellerons par la suite « webmestres débutants » ou « amateurs », auront besoin d'être guidés pas à pas. Pour cela, et comme notre site d'aide s'adresse plus particulièrement aux webmestres débutants, nous leur proposons une démarche personnalisée vers l'accessibilité selon le type de site web qu'ils ont à créer ou à rendre accessible.

Toujours dans l'optique d'assister les webmestres dans leur démarche d'accessibilité, nous avons créé un validateur des normes menant à l'accessibilité. A savoir que notre validateur est capable de vérifier les normes d'accessibilité mais également les normes du langage de description, pré-requis indispensables pour atteindre l'accessibilité. De ce fait le webmestre débutant peut conserver dans toute sa démarche le même validateur.

Une fois les erreurs détectées, il est parfois long et fastidieux de les corriger. Nous avons créé pour cette étape un correcteur de pages web. Cet outil possède deux modes : un mode automatique et un mode manuel. Dans le mode automatique, le correcteur fait lui-même des choix pour modifier la page web. Aucune intervention humaine n'est requise. Cependant de nombreuses erreurs d'accessibilité ne seront pas corrigées efficacement dans ce mode. Dès lors le mode manuel permet quant à lui d'interroger le webmestre, et de modifier la page web selon les informations qu'il aura fournies à notre outil de correction.

Lorsque tous les critères d'accessibilité sont validés, la page est dite « accessible ». Toutefois, comme nous le verrons par la suite, l'utilisabilité d'un site web est également très importante. Par exemple, les plans de sites web peuvent faciliter la navigation de l'internaute et conduire le webmestre à organiser de manière cohérente les pages entres elles. Nous avons alors créé un générateur de plan de site web qui propose d'afficher plusieurs types de plans : le plan réel du site (par l'étude des liens entre les pages) et le plan regroupant les pages de même thématique. Le premier type de plan présente au webmestre une vue globale de son site. Le deuxième type de plan permet au webmestre de visualiser une arborescence réalisable de son site web. Dans ce cas, il a la possibilité de définir un nombre maximal de pages web à chaque niveau de l'arborescence afin d'équilibrer la structure globale. Pour résoudre ce type de problèmes de transformation de graphe en arbre, nous nous sommes inspirés des fourmis qui possèdent les capacités requises dans leur milieu naturel, et avons de ce fait utilisé un algorithme de fourmis artificielles.

Améliorer les interfaces

Les interfaces entre l'internaute et l'ordinateur constituent également un point sensible concernant l'accessibilité des services offerts par internet (web, mail...). Les personnes visuellement handicapées qui connaissent le braille peuvent accéder à Internet *via* une aide technique appelée « clavier braille ». Pour accélérer la lecture avec ce dispositif nous avons réalisé un générateur de braille abrégé, qui pourra être couplé à d'autres outils destinés à faciliter l'accès aux informations contenues dans les pages web. Pour les utilisateurs à mobilité réduite, l'accès à l'information et la lecture ne posent pas réellement de problème. En

revanche la saisie d'information peut être lente et difficile. Sur internet de nombreux formulaires sont présents pour, par exemple, envoyer un mail ou s'inscrire dans le but d'obtenir un service. Pour améliorer la vitesse d'écriture des personnes à mobilité réduite, nous avons élaboré un clavier virtuel s'adaptant, par réorganisation des touches, au type de texte à saisir. La réduction des distances à parcourir entre chaque touche consécutive dans un texte permet ainsi aux personnes handicapées de saisir du texte en minimisant les efforts à fournir. Pour ce faire, nous avons là encore eu recours à des algorithmes de fourmis artificielles.

Améliorer l'interprétation des pages web par le navigateur

Force est de constater que de nombreux sites web ne sont pas accessibles. Même si la loi en France impose l'accessibilité aux sites web publics, cette loi tarde à être mise en application, et (tenant pour exemple des études sur les autres pays européens ayant eux aussi adopté ce type de loi) son respect risque de prendre du temps. De plus de nombreux sites web ne sont pas soumis à cette loi comme par exemple les sites commerciaux. Ils n'ont effectivement aucune obligation d'accessibilité même s'ils pourraient pourtant être très utiles aux personnes handicapées.

Nous avons alors contribué à améliorer l'interprétation des pages web par le navigateur des personnes handicapées pour leur permettre de se servir d'Internet dans de meilleures conditions, même face à un site web non accessible. Pour éviter la relecture d'un contenu identique d'une page à l'autre d'un même site web, nous avons créé un réorganisateur des blocs d'information dans une page web. Notre réorganisateur distingue les différents blocs d'une page web comme le menu, le logo, le titre, les informations pertinentes… L'internaute peut ensuite choisir de n'afficher qu'un seul bloc (et pourquoi pas le lire rapidement avec notre générateur de braille abrégé) ou de choisir un ordre de présentation des blocs pour les pages du site. Le réorganisateur permet d'accélérer l'accès à l'information désirée, pour que, couplé avec le générateur de braille abrégé, l'internaute non-voyant puisse lire rapidement le contenu pertinent.

Remarquons que certains outils destinés à faciliter la conception de sites web accessibles pour les webmestres peuvent également être utiles pour améliorer l'interprétation des pages web par le navigateur de l'internaute. C'est le cas du correcteur de page web. Le webmestre l'utilise essentiellement en mode manuel pour s'en servir en tant qu'assistant effectuant sur ses ordres les modifications (souvent répétitives) dans le code source de la page web. Le mode automatique ne présente que peu d'intérêt pour le webmestre car il n'est consulté pour aucune modification, et donc des erreurs non corrigeables automatiquement subsisteront. En revanche il présente des avantages à l'internaute. En utilisant le correcteur en mode automatique, l'internaute peut accéder sans intervention de sa part, à la page web modifiée. Loin de nous l'idée de rendre une page totalement accessible de manière automatique, mais présenter à l'internaute une page « plus accessible » est déjà un avantage non négligeable. De même le générateur de plan s'adresse au webmestre mais également à l'internaute. Quel que soit le type de plan généré un affichage graphique peut être présenté au webmestre. L'internaute préfèrera un affichage du plan sous forme de liste et cliquable pour l'aider à naviguer. L'utilisateur choisit lui aussi le type de plan à générer : plan réel, par thème, ou

par thème avec nombre maximum de sous-liens fixé. Ce dernier type de plan s'adapte aux capacités de l'internaute. Il peut être utile pour les personnes non-voyantes qui ne peuvent mémoriser trop d'informations de même niveau avant de faire un choix sur la destination, c'est-à-dire le lien à suivre.

La figure 2 représente les principaux outils développés selon le type d'utilisateur à qui ils sont destinés.

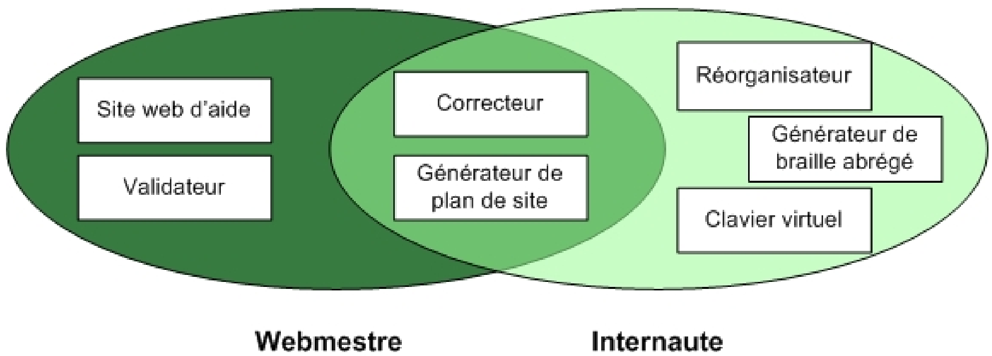


FIG. 2 – Regroupement des outils selon l'utilisateur.

Organisation du document

Ce document est structuré en deux parties. La première s'intéresse aux états de l'art concernant les domaines de l'accessibilité du web que nous avons choisi d'aborder. Etant un domaine de recherches très vaste et en expansion, cette partie ne prétend pas à l'exhaustivité sur l'accessibilité du web dans sa globalité mais couvre uniquement les thèmes que nous avons élus pour nos recherches. La deuxième partie présente quant à elle nos contributions concernant l'accessibilité du web aux personnes handicapées. Nous avons regroupé les différentes entités selon leur type, en trois ensembles (figure 3) qui feront l'objet des chapitres 4, 5 et 6.

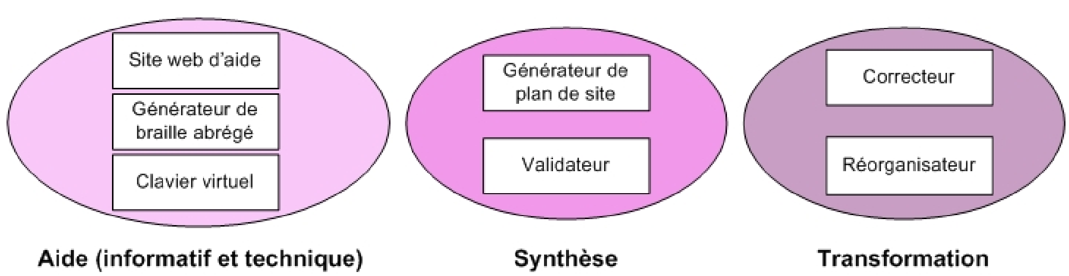


FIG. 3 – Regroupement des outils selon leur rôle.

Aussi, pour mettre de l'ordre dans cette multiplicité de projets et par soucis de clarté, nous proposons ici une rapide présentation des chapitres qui composent notre document :

- le chapitre 1 introduit le problème de l'accessibilité du web, aux besoins des personnes handicapées, aux normes définies dans ce domaine et aux lois mises en place ;
- le chapitre 2 établit des états sur l'art : d'une part le repérage sur un site web grâce au plan de site et d'autre part les systèmes alternatifs d'écriture au clavier traditionnel tels les claviers modifiés et les claviers virtuels ;
- le chapitre 3 présente une méthode de résolution des problèmes combinatoires grâce à l'heuristique ACO (Ant Colony Optimization) ;
- le chapitre 4 décrit trois outils d'aide à l'accessibilité : le premier pour aider les webmestres à rendre leur site web accessible, et les deux autres, destinés quant à eux aux internautes, pour transcrire un texte en braille abrégé et minimiser la fatigue lors de la saisie d'un texte sur un clavier virtuel ;
- le chapitre 5 expose deux outils de synthèse que sont un validateur des normes d'accessibilité ainsi qu'un générateur de plan de site proposant au webmestre plusieurs structures possibles, autres que la structure réelle ;
- le chapitre 6 décrit deux outils de transformation de pages — un correcteur et un réorganisateur — en vue d'améliorer leur accessibilité.

Précisons que la lecture de nos contributions apportées au cours de ce travail peut se faire de manière non-linéaire. Cependant, la lecture antérieure des chapitres de la première partie de ce document permet au lecteur de se faire une idée précise des problématiques abordées dans nos contributions. Les pré-requis à chaque section de la deuxième partie sont indiqués sur la figure 4.

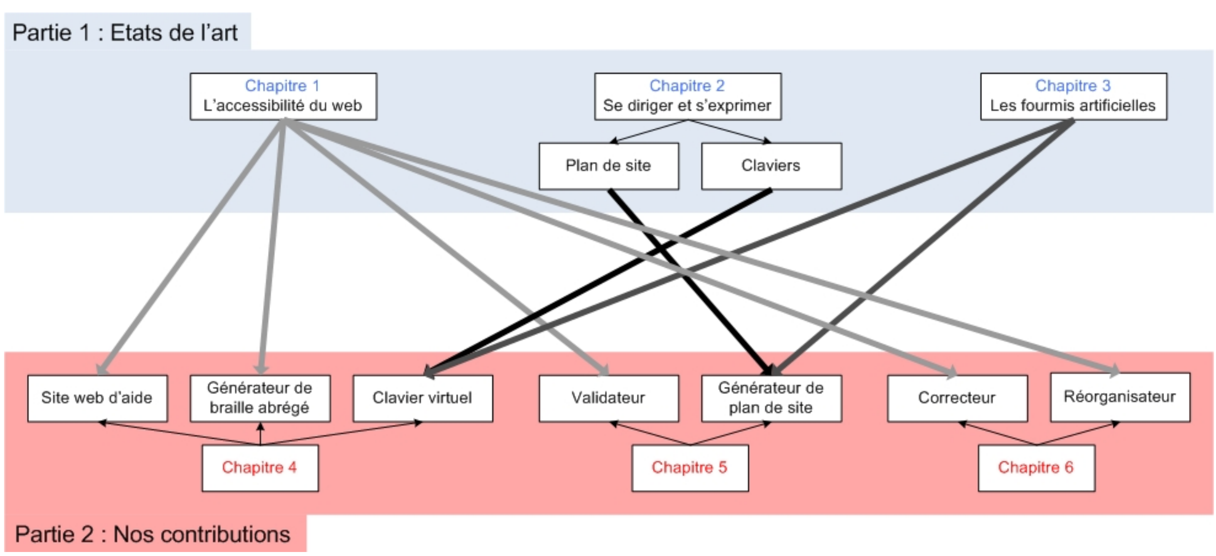


FIG. 4 – Organisation du document en deux parties et relations entre ces dernières.

Une conclusion dresse un bilan sur l'ensemble des travaux présentés dans la seconde partie de ce document et définit les perspectives.

Première partie

Etats de l'art

Chapitre 1

L'accessibilité du web

Dans ce chapitre nous introduisons le problème de l'accessibilité du web aux personnes handicapées visuelles. Ce problème va nous préoccuper dans l'ensemble de ce document. En premier lieu nous allons définir les besoins de ces personnes face à l'informatique, puis plus précisément sur Internet. Et enfin nous établirons les normes à respecter pour leur rendre le web accessible.

1.1 Introduction

Avec l'essor d'Internet, de nombreux services ont été mis en place, tels des services commerciaux (achats en ligne, réservations de billets de train ou d'avion...) [86], et des services administratifs. Outre le fait que ces « e-services » puissent être perçus comme un moyen plus rapide d'obtenir des documents, des renseignements ou même des objets, ils présentent surtout une réelle facilité pour bon nombre de personnes handicapées.

Des démarches administratives qui, par exemple, requièrent un déplacement en mairie — aussi banales ou anodines qu'elles puissent paraître — restent encore difficilement abordables pour les personnes atteintes de cécité. Dans l'idéal, l'utilisation d'Internet devrait s'avérer plus simple et plus rapide pour obtenir le même document. Or dans l'état actuel des choses, il n'en est rien. Effectivement, l'accès à des documents écrits (numériques ou non) n'est pas si simple pour des personnes handicapées visuelles. Même si l'apparition de l'écriture braille permît une réelle avancée en terme d'autonomie des personnes atteintes de cécité, son apprentissage est long et fastidieux, et trop peu de documents sont imprimés en braille. Avec l'avènement des nouvelles technologies numériques, et vu le besoin des personnes handicapées d'utiliser seul un ordinateur, des aides techniques ont été créées. Ainsi grâce notamment à la synthèse vocale, les personnes atteintes de cécité peuvent *« lire »*, ou

plutôt *« écouter »* des documents écrits, et ce sans connaissance du braille. Les personnes handicapées, ayant accès à l'informatique grâce aux aides techniques, devraient par extension pouvoir naviguer sur Internet. Néanmoins pour pouvoir être interprétés correctement *via* les aides techniques, les sites Internet doivent respecter un certain nombre de normes. Actuellement très peu de sites web sont accessibles, c'est pourquoi de récentes lois en Europe imposent aux sites web institutionnels, dans un premier temps, de respecter un niveau minimum d'accessibilité.

Ce chapitre dresse un état de l'art sur l'accessibilité du web. Tout d'abord la section 1.2 définit la notion de « handicap » et présente les aides techniques pour pallier aux incapacités induites par différentes formes de handicap. L'accessibilité du web étant devenue ces dernières années un enjeu majeur, nous mettons en évidence son importance dans la section 1.3 en présentant les recommandations, les labels, et les lois mises en place. Pour finir, nous donnons dans la section 1.4 des moyens concrets pour rendre ou créer un site web accessible.

1.2 Présentation générale du handicap

1.2.1 Définition et chiffres sur le handicap

Classification des handicaps

Le terme « handicap » provient, à l'origine, d'un jeu anglais « hand in cap » consistant à évaluer la valeur des objets et ressemblant à des enchères. La signification du mot handicap a ensuite évolué. Il s'utilise lors des courses hippiques pour comparer la valeur des chevaux. Plus tard, dans les années 1920, le terme handicap est introduit en France, toujours dans le domaine équestre, avec l'idée d'égaliser les chances entre les concurrents, en imposant des désavantages aux meilleurs d'entre eux. Ainsi vînt l'idée d'infériorité associé au terme de handicap (*Dictionnaire historique de la langue française, Le Robert 1992*), qui remplace les termes d'invalidité et d'infirmité, considérés de nos jours comme péjoratifs. Actuellement, le handicap est définit comme un désavantage social résultant d'une déficience ou d'une incapacité qui limite ou interdit l'accomplissement d'un rôle normal. L'utilisation de ce terme reste liée à la notion d'égalisation des chances, puisqu'il exprime une volonté publique de compréhension des différences et d'intégration de tous dans la société.

Pour définir le nombre de personnes handicapées — qui peut varier de 800 000 à plus de 5 millions en France selon la définition retenue du handicap — l'Organisation Mondiale de la Santé (O.M.S.) [230] a aidé en 1980 à la clarification de la notion de handicap en adoptant la classification ICIDH (International Classification of Impairments, Disabilities and Handicaps). Cette classification a été mise au point dans les années 1970 par le Dr. Wood. Elle représente les relations entre maladies et handicaps selon le schéma nommé « séquence de Wood » [94] illustré par la figure 1.1.

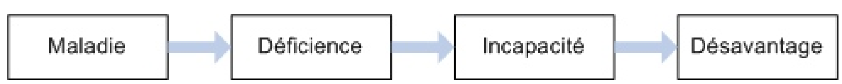


Fig. 1.1 – Séquence de WOOD.

Dans la séquence de Wood, les maladies sont à l'origine de la chaîne. Ce sont des pathologies comprises dans un sens large (accidents, traumatismes moraux ou physiques, conséquences des complications de grossesses...). Les déficiences, qui sont des pertes ou des dysfonctionnements des diverses parties du corps ou du cerveau — on peut parler plus couramment d'invalidité — résultent en général d'une maladie. Ces déficiences engendrent des incapacités, qui correspondent à des difficultés ou des impossibilités de réaliser des actes élémentaires comme se tenir debout. Pour conclure cette séquence, les désavantages désignent les difficultés ou impossibilités que rencontre une personne pour remplir les rôles sociaux auxquels elle peut aspirer, ou que la société attend d'elle.

La classification ICIDH fut traduite en français en 1988 sous le titre de « Classification Internationale des Handicaps : déficiences, incapacités, désavantages » (CIH). Le titre en français n'est pas la traduction exacte du titre en anglais. Pour les anglophones, le terme

« handicap » correspond en effet au dernier maillon de la séquence de Wood, alors qu'en France il regroupe à la fois les domaines de déficience, d'incapacité et de désavantage. Par la suite, en 2002, la CIH — qui était une approche médicale du handicap — évolue pour donner la CIH - 2 [100], également appelée la Classification Internationale du Fonctionnement, du handicap et de la santé (CIF) [232], et prenant désormais en compte les facteurs environnementaux et sociaux. Le schéma de cette classification illustré par la figure 1.2 s'oriente donc vers l'activité des personnes handicapées.

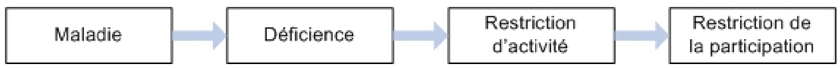


FIG. 1.2 – Séquence de la CIF.

Les classifications des handicaps distinguent les handicaps mentaux et les handicaps physiques.

Concernant les handicaps mentaux, deux types peuvent être identifiés. D'une part, on observe les handicaps mentaux résultant de déficiences intellectuelles, autrefois repérés par des vocables comme « arriération mentale » ou « débilité mentale ». Les différentes batteries de tests de QI sont des outils classiques pour leur appréciation, mais ils sont très réducteurs et insuffisants pour décrire les difficultés réelles des personnes. On peut discerner dans cette catégorie complexe des déficiences plus spécifiques de certaines fonctions cognitives, comme celles du langage ou d'autres troubles spécifiques d'apprentissage. D'autre part, on distingue les handicaps mentaux résultant de déficiences psychiques. Ils concernent les troubles du fonctionnement de l'appareil psychique et influent donc principalement sur les sphères de la vie relationnelle, de la communication, du comportement...

De même les handicaps physiques se répartissent en deux grandes catégories. La première concerne les handicaps moteurs, ou déficiences motrices. Ce handicap représente l'image même du handicap dans l'imaginaire collectif : le pictogramme symbolisant le handicap est la représentation d'une personne en fauteuil roulant. Ce sont donc des handicaps en général visibles mais leurs expressions et leurs conséquences sont très variables. La seconde catégorie correspond aux handicaps sensoriels. L'importance du déficit sensoriel amène à créer des sous-catégories. D'un côté, la surdité, ou déficience auditive, est une « diminution unilatérale ou bilatérale de l'ouïe, quels qu'en soient le degré et l'origine » (*Dictionnaire de Médecine, Flammarion, 1975*). La surdité entraîne une impossibilité à capter correctement les informations sonores ainsi qu'une impossibilité à y répondre correctement oralement par manque de contrôle des émissions sonores. D'un autre côté, la cécité est une absence totale de la vue. Si le nombre de personnes atteintes de cécité en France tend à diminuer grâce aux progrès de la médecine, le nombre de personnes malvoyantes est en augmentation, ce qui est dû principalement au vieillissement de la population.

Les chiffres du handicap en France

Selon l'OMS, le handicap regroupe les domaines des déficiences, des incapacités et des désavantages. Comme l'atteinte peut être minime ou conséquente, un seuil est nécessaire pour évaluer le nombre de personnes handicapées. Pour pouvoir dénombrer cette population, des indicateurs chargés de synthétiser l'information ont été mis en place. Le plus ancien d'entre eux est celui de KATZ [22], qui a été bâti à partir de la sélection des activités de la vie quotidienne considérées comme essentielles (se laver, s'habiller, aller aux toilettes, entrer et sortir de son lit ou s'asseoir, se lever d'une chaise ...). Celui de Colvez [45], appliqué dans les années 1980 au secteur des personnes âgées, est un indicateur de mobilité dont la diffusion est désormais plus large, notamment depuis sa reprise par l'enquête « Handicaps, Incapacités, Dépendance » (HID) [150]. Il isole quatre groupes de personnes : celles confinées au lit ou au fauteuil, celles ayant besoin d'aide pour la toilette ou l'habillage, celles ayant besoin d'aide pour sortir, et les autres cas. Cette enquête HID réalisée par l'INSEE (Institut National de la Statistique et des Etudes Economiques) et l'INSERM (Institut Nationale de la Santé et de la Recherche Médicale) entre 1996 et 2001, est la première enquête nationale sur le handicap et la dépendance. Elle a pour objectif de compter et de décrire la population concernée et est composée pour cela de quatre vagues de collectes [149]. La première, menée au dernier trimestre 1998, a concerné les personnes vivant en institutions. Elle a permis de recueillir les réponses de 14 600 personnes. Fin 1999, le même questionnaire a été adressé à un échantillon de 16 900 personnes, vivant en domicile ordinaire. Un second passage a eu lieu fin 2000 et début 2001 auprès des mêmes personnes, afin d'évaluer les évolutions [96]. L'enquête HID portait sur les domaines suivants : l'entretien personnel, la mobilité, la souplesse, l'orientation dans le temps et la communication, la vue et l'audition. Des documents de travail synthétisent les résultats de cette enquête concernant les personnes en institution [24]. De manière générale, cette enquête a montré que les problèmes de souplesse sont les plus fréquents. Plus précisement, chez les hommes, les problèmes de vue ou d'audition viennent en deuxième position, suivis des problèmes de mobilité, alors qu'il se produit l'inverse chez les femmes. Celles-ci souffrent davantage de déficiences motrices tandis que les hommes souffrent plus de déficiences sensorielles [21].

D'un côté, les déficiences motrices touchent au total 8 millions de personnes en France, regroupant aussi bien des atteintes modérées voire légères, comme le rhumatisme et l'arthrose, que les grandes paralysies telles la tétraplégie, la paraplégie et l'hémiplégie. Ces trois atteintes les plus sévères concernent moins de 9% des déficients moteurs, soit moins de 1% de la population. Globalement, les déficiences motrices affectent 1% des jeunes enfants, plus de la moitié des octogénaires et deux tiers des nonagénaires [95].

D'un autre côté, parmi les déficiences sensorielles, 2 millions de personnes présentent une déficience auditive légère en France, 1 250 000 une déficience auditive moyenne, 350 000 une déficience auditive sévère et 110 000 une déficience auditive profonde. Précisons que 2 400 000 personnes âgées de plus de 60 ans sont atteintes de déficience auditive. D'autre part, environ 10% de la population connaît des difficultés visuelles à des degrés divers. On estime à environ 70 000 le nombre d'aveugles en France, et entre 1,5 et 3 millions de malvoyants.

Classification des handicaps visuels

Le terme cécité vient du mot latin « caecus », signifiant aveugle. L'aveugle, au sens strict, est celui qui est privé de ses yeux (ab oculis), celui qui est privé de la vue. Pour estimer l'importance de la déficience visuelle d'un individu, des mesures sont effectuées sur son acuité visuelle et sur son champ visuel. D'un côté, l'acuité visuelle est une mesure d'angle entre l'œil et deux points situés à 5 mètres de lui (distance considérée comme l'infini optique). Se rapprocher pour mieux voir les deux points équivaut à agrandir l'angle et nécessite de ce fait une acuité visuelle moins élevée. L'examen permettant de dépister des troubles de la vue doit être composé d'une part du test sur l'acuité visuelle de loin qui se mesure à l'aide de l'échelle de Monoyer, laquelle comporte 10 lignes de lettres correspondant chacune à 1/10 d'acuité visuelle. Le patient doit se placer à 3 ou 5 mètres selon l'échelle utilisée. D'autre part, l'examen doit également tester l'acuité visuelle de près qui se mesure à l'aide de l'échelle à main de Parinaud — existant également pour enfants et illettrés. D'un autre côté, le champ visuel est la portion de l'espace perçu par un œil maintenu immobile. Le champ visuel périphérique permet de percevoir les mouvements et la pénombre alors que le champ visuel central permet de voir les formes et les couleurs.

En se basant sur l'acuité et le champ visuel pour définir la vision d'un individu, la Classification statistique Internationale des Maladies et des problèmes de santé connexes, dixième révision (CIM-10)[1] [99], permet de classer les déficiences visuelles comme le montre le tableau 1.1. A titre d'information, nous pouvons dire qu'une personne ayant une acuité visuelle de 1/60 est capable de compter sur ses doigts à un mètre. Cette classification de l'OMS sur la déficience visuelle est basée sur l'acuité visuelle de loin, du meilleur œil après correction.

Catégories de Déficiences possibles (CIM-10)	Acuité visuelle du meilleur oeil avec la meilleure correction	
	Minimum (acuité supérieure ou égale à)	Maximum (acuité inférieure à)
1	1/10	3/10
2	1/20	1/10
3	1/60	1/20
4	Perception de la lumière	1/60
5	Pas de perception de lumière	
6	Indéterminée ou sans précision	

TAB. 1.1 – Classification des atteintes visuelles définie dans la CIM-10 (OMS 1993).

Parmi ces catégories, il est possible de différencier les deux notions classiques qui sont la cécité et l'amblyopie. La cécité correspond à une acuité visuelle du meilleur œil avec correction, au maximum inférieure à 1/20 (catégories de déficience visuelle 3, 4 et 5 de la CIM-10), ou à un champ visuel inférieur à 10° quelle que soit l'acuité visuelle. L'amblyopie (ou malvoyance), correspond à une acuité visuelle du meilleur œil avec correction supérieure

[1] La CIM-10 est la traduction française de l'ICD-10, International Statistical Classification of Diseases and Related Health Problems.

ou égale à 1/20 mais inférieure à 3/10 (catégories de déficience visuelle 1 et 2 de la CIM-10), ou à un champ visuel compris entre 10° et 20° pour chaque œil.

Le tableau 1.2 permet d'établir une correspondance entre les différents termes, et les catégories de déficiences visuelles de la CIH-10. Nous distinguons ici différents types de cécité [1].

	ou	
	Catégorie CIM-10	Champs visuel
amblyopie	1 ou 2	10° < ... < 20°
cécité professionnelle	3	< 10°
quasi-cécité	4	
cécité complète	5	

Tab. 1.2 – Correspondance entre le vocabulaire et les déficiences.

En ce qui concerne la malvoyance, on peut distinguer plusieurs atteintes :

- la vision centrale
  Dans ce cas, l'atteinte visuelle concerne la partie centrale de la rétine. Les cellules regroupées à cet endroit permettent non seulement la vision des formes et des couleurs, mais ont surtout un très grand pouvoir discriminant. Ce sont elles qui rendent possible la vision des détails. Avoir une atteinte de cette partie de la rétine implique donc une réduction importante de l'acuité visuelle. Par conséquent, les personnes ayant ce type d'atteinte auront des difficultés ou seront dans l'impossibilité de lire de près des caractères de taille normale (de type dactylographique, d'imprimerie ou de journaux), ou de plus loin des caractères un peu plus grands (le nom d'une rue, le numéro de quai d'un train...). Finalement, il sera nécessaire d'utiliser des moyens de grossissement.
- la vision périphérique
  Les sujets n'ont plus de perception visuelle possible ou de qualité suffisante, autour du point de fixation. Au contraire, leur champ visuel se rétrécit jusqu'à devenir tubulaire. Ils conservent une acuité correcte, mais ils ne voient que ce qu'ils fixent et plus rien autour. Ces sujets conservent des capacités de lecture mais il faut, pour lire efficacement, que la taille des caractères soit suffisamment réduite pour ne pas dépasser la largeur de leur champ visuel. A l'opposé de l'idée communément admise, grossir un texte consiste pour eux à accroître d'autant leurs difficultés.
- la vision floue
  Dans ce cas, la vision des sujets ressemble à celle que l'on peut avoir à travers un verre dépoli. A savoir que la lumière passe toujours mais dans un milieu qui, par son opacité, la diffuse. Dès lors, la source lumineuse, au lieu de se réfléchir précisément sur la rétine, se diffuse et ses contours s'atténuent. Les lettres et les petits détails peuvent être mal perçus, de même les grandes formes peuvent paraître estompées ou déformées selon l'éclairage et les reflets. Pour ces patients, une forte luminosité est « aveuglante » alors que pour un voyant elle est supportée voire adaptée. On dit qu'ils sont photophobes, ce qui signifie que leur tolérance à la lumière est inférieure à la normale. Des moyens de grossissement couplés à un éclairage adéquat peuvent aider ces personnes à lire.

Les chiffres du handicap visuel en France

L'OMS estime à environ 70 000 le nombre d'aveugles en France répartis comme suit : 20 000 enfants et adolescents et 50 000 adultes. La population atteinte de cécité totale est en régression grâce aux progrès thérapeutiques mais dans le même temps, ceux-ci concourent à l'augmentation du nombre de malvoyants (entre 1,5 et 3 millions de malvoyants). Selon l'enquête HID (Handicaps, Incapacités, Dépendance), 3,1 millions de personnes déclarent une atteinte visuelle, parmi lesquelles 55 000 personnes souffrent d'une cécité complète (avec éventuellement perception de la lumière) et 225 000 souffrent d'une cécité partielle (avec distinction de silhouette) [95]. On peut remarquer des similitudes entre les chiffres donnés par l'estimation de l'OMS et par l'enquête HID bien qu'il reste difficile d'évaluer exactement le nombre d'handicapés visuels. En effet la comparaison de ces chiffres nécessiterait de définir dans chaque étude le seuil au delà duquel une personne est considérée comme handicapée. De plus de nombreuses personnes ayant des incapacités légères ne se considèrent pas comme étant handicapées et ne signalent donc pas leur déficience.

Une étude menée par l'OMS en 2002 [231] a permis d'établir une distribution des atteintes visuelles par zone géographique (tableau 1.3). Nous pouvons remarquer que les atteintes visuelles ne sont donc pas uniformément réparties dans le monde. Les pays en développement enregistrent en effet plus de 90% des cas.

	Afrique	Amérique	Méditerranée Orientale	Europe	Asie du Sud Est	Pacifique Occidental	Total
Population	672.2	852.6	502.8	877.9	1590.8	1717.5	6213.9
Nombre d'aveugles	6.8	2.4	4.1	2.7	11.6	9.3	36.9
% sur le nombre total d'aveugles	18%	7%	11%	7%	32%	25%	100%
Nombre de mal voyants	20	13.1	12.4	12.8	33.5	32.5	124.3
Nombre de déficients visuels	26.8	15.5	16.5	15.5	45.1	41.8	161.2

TAB. 1.3 – Estimation mondiale des atteintes visuelles par Région OMS (en millions), 2002.

1.2.2 Les handicapés visuels et l'accès à la culture écrite

Les premiers systèmes d'écriture en relief

Pendant des siècles, seuls les aveugles fortunés avaient accès à la culture écrite à l'aide de personnes voyantes leur faisant la lecture. Les quelques essais d'écriture pour aveugles étaient tous basés sur le même principe : l'alphabet latin était reproduit sous forme de lettres en relief, sculptées ou gravées dans le bois, la cire ou le métal.

C'est au XVIIème siècle que François Lana Terzi (1631-1687) propose en 1670 un système d'écriture utilisant des symboles au lieu de l'alphabet latin. Il conçoit également un procédé d'impression en relief sur du papier épais. La figure 1.3 illustre ce système d'écriture en présentant le modèle ainsi qu'un exemple. L'encadrement représente l'emplacement dans le modèle et désigne alors deux ou trois lettres. Puis le nombre de points permet d'identifier le caractère.

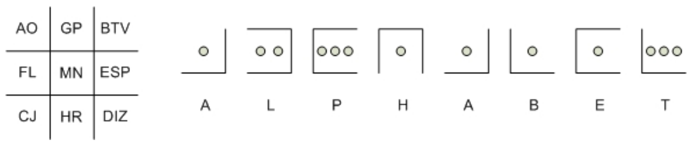


FIG. 1.3 – Alphabet de François Lana Terzi (1670).

Au siècle suivant, Valentin Haüy (1745-1822) fonde la première école pour aveugles gratuite et ouverte à tous — laquelle deviendra par la suite l'Institution Royale de Jeunes Aveugles (actuellement l'Institut National des Jeunes Aveugles). Il imprime pour ses élèves les premiers livres en relief. Dans cette école, les élèves aveugles apprennent la lecture et l'écriture avec des caractères romains en relief. Toutefois, les caractères romains sont difficiles à reconnaître au toucher ce qui implique que la lecture est extrêmement lente.

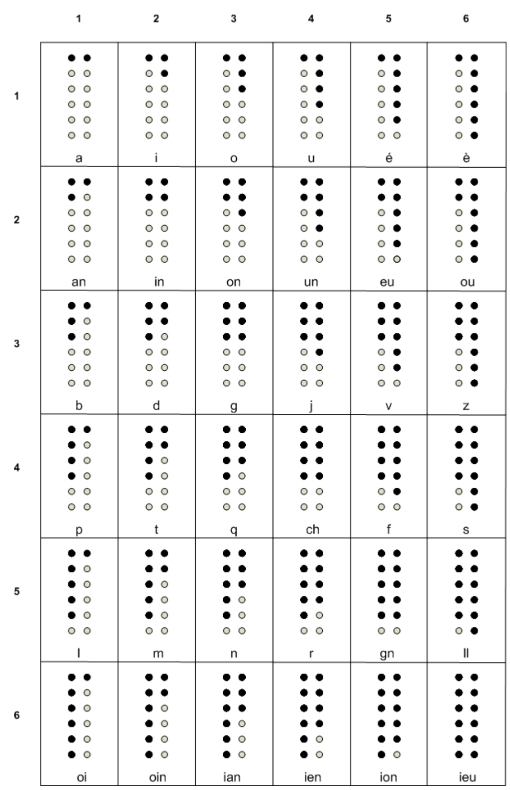


FIG. 1.4 – Système phonétique de Charles Barbier (1821).

Pour pallier la lenteur de cette méthode, Charles Barbier (1767-1841) invente un nouveau système appelé « sonographie ». Ce système, illustré par la figure 1.4, représente des sons à l'aide de 12 points en relief. Il sera testé par les élèves de l'Institution Royale de Jeunes Aveugles dès 1821. Comparés aux lettres en relief, les points sont beaucoup plus faciles à lire et à écrire ; cependant cette écriture est phonétique et de ce fait ne respecte pas l'orthographe. Elle ne permet pas non plus de transcrire les signes de ponctuation, les chiffres et les symboles mathématiques. Sans oublier que les caractères peuvent atteindre jusqu'à douze points, ce qui correspond à une hauteur trop élevée pour être lue immédiatement par le doigt.

Louis Braille (1809-1852), élève à l'Institution Royale de Jeunes Aveugles, a connu les deux systèmes d'apprentissage (l'alphabet de Terzi et le système phonétique de Barbier). Afin de mieux répondre aux besoins des aveugles, il décide de perfectionner le système de Barbier. C'est ainsi qu'en 1829 paraît le premier exposé de la méthode Louis Braille. Il s'agit d'un procédé pour écrire du texte et de la musique au moyen de points. En 1837, une seconde édition du procédé donne la version définitive de l'alphabet, illustrée par la figure 1.5. Les caractères braille sont formés de combinaisons de six points, disposés en deux colonnes de trois points.

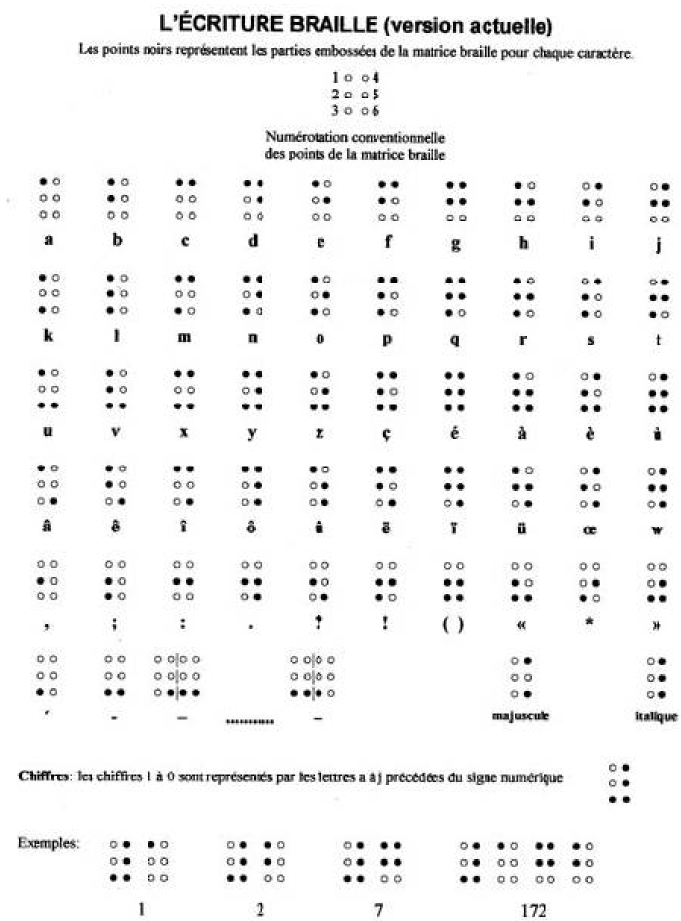


FIG. 1.5 – Alphabet braille (1837).

La méthode actuelle : le braille

Il existe désormais plusieurs types de braille : le braille intégral, le braille abrégé, le braille musical et le braille mathématique et scientifique :

1. Le braille intégral
   Le braille intégral [71] fait correspondre à chaque lettre ordinaire une lettre braille. La lettre braille est composée de 6 points, disposés en deux colonnes de trois points ce qui offre 2^6 possibilités soit 64 combinaisons. L'alphabet braille français est composé de 26 lettres et 14 signes (13 lettres accentuées et le « ç »). En plus de ces 40 combinaisons, 10 sont utilisées pour les signes de ponctuation, et les 14 restants, appelés caractères « hors séries », sont utilisés essentiellement pour le braille abrégé. Précisons que certains caractères « hors séries » ont une valeur en braille intégral. Il existe par exemple une combinaison représentant le signe majuscule et représentant en majuscule toute lettre devant laquelle il est placé. Pour représenter une lettre en majuscule il faut donc utiliser 2 signes braille. La figure 1.6 illustre les deux caractères nécessaires à l'écriture d'un « A » majuscule.

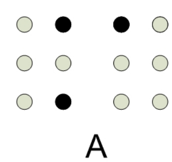
   

   Fig. 1.6 – Représentation en braille de la lettre A.

   Le braille intégral est donc un code simple mais qui prend beaucoup de place. Une page de braille contient effectivement sous un format $(21 * 29,7)$ au maximum 27 lignes de 30 caractères — ceci étant dû à la particularité du signe braille qui possède une taille irréductible. Il est vrai qu'un livre de braille intégral peut représenter ainsi jusqu'à 6 fois plus de volume que le même document en écriture ordinaire. Par conséquent, l'écriture des textes en braille intégral exige une grande quantité de papier et des coûts de production élevés, puisque les caractères braille sont plus larges que leurs équivalents en « noir ». A cela s'ajoute un autre inconvénient du braille : le rythme de lecture en braille s'avère plus lent que celui atteint avec les yeux.

2. Le braille abrégé
   Le braille abrégé [72] permet de réduire la taille du texte et d'effectuer une lecture plus rapide. L'écriture en braille abrégé repose sur un système de symboles et de contractions. Les symboles sont des mots pré-abrégés qui ont une transcription définie (un ou deux signes braille). Quant aux contractions, elles sont des abrégés de mots ou de groupes de lettres apparaissant dans des conditions particulières. Ainsi plusieurs groupes de lettres différentes peuvent être abrégées de la même manière, selon leur position dans le mot (devant une consonne, en début de mot, pas en fin de mot, devant une voyelle...). Un texte en braille abrégé est donc composé de mots contenant une ou plusieurs contractions, de symboles, et de mots en texte intégral. Aussi nous comprenons mieux que l'avantage du braille abrégé est la réduction de 20% environ

des ouvrages écrits en braille. Remarquons également que le braille abrégé, qui obéit à des règles phonétiques et orthographiques (par exemple, il ne faut pas contracter un groupe de lettres définissant plusieurs syllabes), est spécifique à chaque langue.

Transcrire un texte en braille abrégé consiste à découper un texte en mots, à les soumettre à un traitement de locutions et ensuite à vérifier les règles de compression :

– Les locutions
  En français, les locutions sont formées soit de mots séparés, soit de mots unis par un trait d'union ou par une apostrophe. Les règles d'abréviation des locutions précisent qu'un trait d'union et une apostrophe doivent être retranscrits dans la locution abrégée. Les locutions constituées de plusieurs mots non reliés doivent être unies lors de leur transcription en braille abrégé à l'aide du signe italique en abrégé. La figure 1.7 représente un exemple d'abréviation d'une locution non unie. Par exemple, la locution « par conséquent » est abrégée par 3 caractères braille.

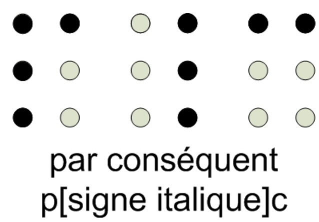


FIG. 1.7 – Abréviation d'une locution.

– Les symboles
  Certains mots peuvent être abrégés par des symboles. En général ces derniers ne peuvent pas être utilisés pour contracter une partie d'un mot. Ils peuvent néanmoins être employés au pluriel en ajoutant un « s » à la contraction, au féminin en ajoutant un « e », ou au féminin pluriel en ajoutant « û ». La figure 1.8 illustre l'abréviation du mot « petit ».

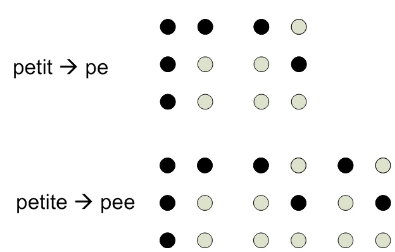


FIG. 1.8 – Symbole de « petit » et « petite ».

– Les contractions
  Certains mots peuvent être abrégés à l'aide de contractions. Une contraction permet d'abréger une chaîne de caractères entrant dans la composition d'un mot. Certaines contractions sont toujours applicables, par exemple la chaîne « our » se contracte en « ç » quelle que soit sa position dans le mot. En revanche, d'autres contractions ne peuvent être réalisées que si elles vérifient quelques critères. Certaines chaînes de

caractères ne peuvent par exemple être contractées qu'en début de mot, qu'en fin de mot, suivi d'une consonne, ou d'une voyelle. Ces restrictions sont très importantes car il existe en braille abrégé des contractions représentant des chaînes de caractères différentes selon leur position dans le mot. Par exemple la contraction « â » correspond à la chaîne « ation » si celle-ci est en fin de mot ou bien à la chaîne « fr » si celle-ci est suivie d'une voyelle.
De plus, une contraction ne peut se faire que si la chaîne de caractères à contracter ne chevauche pas deux syllabes. Il est donc nécessaire d'être capable de détecter les différentes syllabes d'un mot et de vérifier ensuite que la chaîne de caractères dont on cherche une contraction appartient à une seule syllabe.
   - Les mots en intégral
     Dans un texte en braille abrégé on trouve également des mots en intégral. Ils ne possèdent pas de symboles et aucune contraction n'a été trouvée. De plus les noms propres ainsi que les sigles doivent rester en intégral.

3. Le braille mathématique
   Le braille mathématique est différent d'un pays à un autre. En France il a été mis au point en 1922 par deux professeurs de mathématiques : Messieurs Antoine et Bourguignon. Le document contenant la transcription de l'ensemble des notations mathématiques paraît en 1971 en braille et en 1973 « en noir ». Une révision de ce code est appliquée depuis septembre 2001 [46] pour tenir compte des évolutions des notations en braille littéraire et pour faciliter la transcription automatique à l'aide de logiciels. Par exemple, il était logique que les parenthèses mathématiques soient représentées par des signes identiques aux parenthèses utilisées dans le système braille actuel. Ce changement a entraîné notamment la modification de la représentation du signe « multiplié », qui aurait fait double emploi avec la parenthèse gauche.

   Alors qu'en braille littéraire l'information est linaire, les équations mathématiques sont quant à elles composées d'éléments « en 2 dimensions ». Par exemple une fraction ou une puissance sont des éléments bi-linéaires. Il est possible d'écrire toute équation de manière linéaire afin qu'elle puisse être lue par des non-voyants. Cependant, dans le cas d'équation compliquée, l'écriture linéaire devient rapidement complexe. La codification braille mathématique a donc été mise en place pour représenter les formules mathématiques de manière spatiale. De plus cette codification permet de transcrire les symboles purement mathématiques tels que « $+$ », « $=$ », « $\in$ », « $\Re$ ». L'alphabet braille n'étant composé que de 64 caractères, ces symboles mathématiques sont représentés à l'aide de caractères braille ayant déjà une signification en braille littéraire. Par exemple, le « $+$ » et le « ! » correspondent au même caractère braille. Pour éviter de telles ambiguïtés, toute formule mathématique doit être précédée par les caractères « '. » (c'est-à-dire une apostrophe suivie d'un point) et se termine au premier espace ou retour à la ligne.

4. Le braille musical
   A l'instar des notations mathématiques, les partitions de musique utilisent une représentation spatiale. Comme l'illustre la figure 1.9, les 7 notes de musique sont chacune représentées par un symbole braille (les lettres d, e, f, g, h, i et j). Les points 1, 2, 4 et

5 indiquent le nom de la note et les points 3 et 6 précisent la durée. En l'absence des points 3 et 6, cette note est considérée comme une croche ; elle devient une noire si on ajoute le point 6, une blanche avec le point 3 ou une ronde en ajoutant au symbole de la note les points 3 et 6 (figure 1.10). Les 5 octaves sont également représentés par un symbole braille et doivent précéder toute note changeant d'octave par rapport à la note précédente.

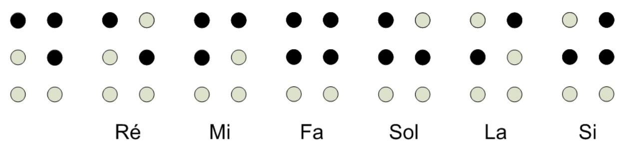


FIG. 1.9 – Représentation des 7 notes de la gamme en braille musical.

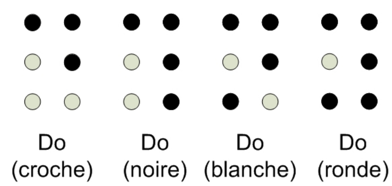


FIG. 1.10 – Représentation des valeurs de la note Do en braille musical.

Il existe désormais des magazines spécialisés pour les aveugles, comme par exemple Carrefour braille[2], publié pour la première fois en 1969 et comptant aujourd'hui 600 abonnés (100 pour le braille intégral et 500 pour le braille abrégé). Des centres de documentation permettent de regrouper tout document et livre braille. Bien souvent ils possèdent également un service de transcription qui propose de convertir tout texte imprimé en braille. On peut, entre autres, citer le CRDA (Centre de Ressources Documentaire et d'Adaptation pour Aveugles et Amblyopes)[3] et le centre CTEB (Centre de Transcription et d'Edition en Braille)[4].

Pour transcrire un texte imprimé en noir (c'est-à-dire avec de l'encre) en un texte imprimé en braille, plusieurs services interviennent. Les Centres de Transcription Spécialisés (CTS) adaptent les fichiers numériques, obtenus par numérisation ou auprès des éditeurs (lesquels détiennent les sources originales de l'œuvre). Ils les transforment dans un format spécifique permettant leur impression ou leur consultation sur ordinateur. Puis les Centres d'Impression Spécialisés (CIS) impriment en gros caractères ou embossent en braille les fichiers adaptés par les CTS.

Pour les personnes aveugles maîtrisant la lecture en braille, ce système est un outil très pratique et a longtemps été le seul moyen de lire des documents. Cependant, très peu de personnes savent lire le braille car son apprentissage est très long. Bien souvent, les personnes

[2]`http://www.inlb.qc.ca/publications/carrefourbraille.aspx` - dernier accès le 23/11/2006
[3]`http://www.iphv.fr/crda.htm` - dernier accès le 23/11/2006
[4]`http://www.irit.fr/ACTIVITES/TOBIA/tobia.html#cteb` - dernier accès le 23/11/2006

devenues aveugles avec les années (ce qui correspond à la plus grande partie des aveugles en France) ne savent pas lire le braille. L'informatique peut alors leur permettre de lire des documents grâce à des aides techniques qui ne nécessitent pas de connaître le braille.

1.2.3 Les handicapés visuels et l'informatique

Pour convertir un texte en noir en un texte imprimé en braille, les CTS ont besoin de logiciels de conversion, et les CIS doivent posséder des imprimantes spécialisées. Les premiers programmes informatiques de transcription de textes en braille sont apparus en 1950. Actuellement, parmi les plus connus on peut citer :

- DBTWIN [185] (Duxbury Braille Translator for Windows) de la société américaine Duxbury (et l'association AVH pour la version française), winbraille [152] et BrailleStar [204] de l'Institut National des Jeunes Aveugles (INJA) pour la transcription de textes en braille intégral ou abrégé ;
- Bramanet [222], financé par le ministère de l'Education Nationale et permettant de convertir un fichier saisi par une personne voyante à l'aide de l'éditeur d'équations de Microsoft Word en texte pré-codé — qui sera ensuite converti en braille par DBTWIN —, pour la transcription en braille mathématique ;
- BME [203] (Braille Music Editor),MFB [75] (Music for the Blind), et Goodfeel [220] pour le braille musical.

Ajoutons que le braille est fondé sur un code binaire (présence ou absence de points) et s'adapte donc parfaitement à l'informatique. Force est de constater que les premières barrettes tactiles en braille qui voient le jour dans les années 1980, permettront aux aveugles d'utiliser la plupart des applications informatiques mettant en œuvre du texte et également de naviguer sur Internet. L'actuelle barrette braille convertit les différents caractères à l'écran en caractères braille grâce à des programmes adéquats et à un système de picots rétractables.

Les aides techniques

Les adaptations techniques (notamment la barrette braille), jouent un rôle essentiel dans le maintien à domicile et l'autonomie des personnes aveugles. L'emploi d'aides techniques permet un accès plus aisé à l'information écrite, imprimée ou à l'informatique. La classification internationale des aides techniques pour personnes handicapées définit les aides techniques comme : *« tout produit, instrument, équipement ou système technique utilisé par une personne handicapée, fabriqué spécialement ou existant sur le marché, destiné à prévenir, compenser, soulager ou neutraliser la déficience, l'incapacité ou le handicap ».*

Les principales aides techniques pour les personnes aveugles et malvoyantes sont les suivantes.

Un logiciel « lecteur d'écran » convertit les données apparaissant à l'écran de l'ordinateur en format compatible avec la barrette braille et/ou le synthétiseur vocal. Les deux logiciels les plus utilisés sont JAWS [195], distribué par Visuaide, et VIRGO [166], distribué par

Baumfrance. Ils proposent différentes méthodes de reproduction, à savoir le mode logique et le mode ligne. Le premier reproduit l'information qui est présente dans le focus — et complète éventuellement cette information par de l'information se trouvant à proximité comme le libellé du champ dans lequel se trouve le focus. Dans le mode ligne, toute l'information présente sur la ligne où se trouve le focus est affichée. De ce fait, ce mode affiche tout ce qui se trouve sur la ligne courante même si ces informations n'ont rien à voir entre elle. Pour pallier ce problème il est possible d'utiliser la représentation spatiale en mode ligne qui permet de montrer — par exemple en laissant des espaces entre les informations sur la barrette braille — que les informations ne sont pas directement liées.

La synthèse vocale est un système qui transmet les données apparaissant à l'écran de l'ordinateur vers un synthétiseur de telle sorte qu'elles soient émises *via* un haut-parleur. La synthèse vocale comporte une partie matérielle (la carte son et le haut-parleur) et une partie logicielle (les programmes « lecteur d'écran » et les voix des différentes langues). Elle est destinée tant aux personnes malvoyantes — pour qui la lecture à l'écran est trop difficile, lente, fatigante, malgré un système d'agrandissement —, qu'aux personnes qui ne lisent pas à l'écran et qui ne pratiquent pas le braille de manière courante, et aussi aux personnes aveugles qui travaillent avec une barrette braille en combinaison avec une synthèse vocale.

La barrette braille — aussi appelée plage braille, afficheur braille, ou encore écran braille — est un dispositif qui affiche en braille les données reçues du lecteur d'écran. La barrette est composée d'une suite de cellules insérées dans un support électronique. Chaque cellule peut faire apparaître les huit points d'un caractère braille grâce à des picots rétractables. Employée en informatique par les personnes aveugles pour avoir accès aux informations de l'écran, la barrette braille est un périphérique qui remplace le moniteur. Elle est souvent placée à l'avant du clavier dactylographique.

Le preneur de notes se substitue au papier et au stylo de la personne bien voyante prendre des notes. Les données sont sauvegardées dans la mémoire du preneur de notes pour être consultées à nouveau. Cet appareil étant connectable à un ordinateur, les données peuvent être retravaillées. Par la suite les notes peuvent être rechargées de l'ordinateur vers le preneur de notes. Remarquons que cet appareil est couramment utilisé dans le cadre de la scolarité ou d'un emploi qui nécessite de fréquents déplacements.

Les personnes handicapées visuelles peuvent également utiliser une embosseuse pour imprimer les textes d'un fichier informatique en écriture braille.

Les personnes aveugles et gravement malvoyantes accèdent à la lecture de documents imprimés ou dactylographiés grâce à un scanner avec la reconnaissance de caractères (OCR, optical character recognition). Le scanner photographie le document de manière électronique. Le programme de reconnaissance de caractères le transforme en texte pour le sauvegarder dans la mémoire de l'ordinateur. La personne peut en prendre connaissance *via* un logiciel d'agrandissement, une synthèse vocale ou une barrette braille. L'éventuelle modification des données est alors possible, comme pour tout fichier informatique.

Il existe également des aides techniques spécifiques aux personnes malvoyantes, comme la vidéo-loupe et le logiciel d'agrandissement.

D'une part, la vidéo loupe est un appareil constitué d'une caméra miniature, d'un plateau mobile et d'un moniteur. Le document est posé sur le plateau, retransmis par la caméra et son image apparaît agrandie sur l'écran. Ce système facilite la lecture et l'écriture à un grand nombre de personnes malvoyantes.

D'autre part, le logiciel d'agrandissement permet d'agrandir toutes les informations qui apparaissent à l'écran suivant les besoins de la personne malvoyante. Le même agrandissement s'applique tant aux textes qu'aux icônes, menus, boîtes de dialogues, etc. Le logiciel libre Another Lens [168], en anglais, permet d'agrandir jusqu'à 8 fois la zone où se trouve le curseur de la souris. L'agrandissement est visible dans une fenêtre qu'il est possible de redimensionner. Un autre logiciel libre, Lens [157], permet un agrandissement jusqu'à 64 fois et la langue est réglable. Remarquons également qu'un moniteur de grande taille est parfois suffisant pour que certaines personnes malvoyantes puissent lire l'écran de leur ordinateur. Certaines personnes malvoyantes combine d'ailleurs un logiciel d'agrandissement avec un grand écran.

Avec l'utilisation de l'informatique pour les personnes aveugles, un nouveau système de codage a vu le jour : il s'agit du braille informatique. Une cellule braille classique est composée de 6 picots ce qui permet de représenter 64 caractères. Les majuscules doivent être représentées en 2 symboles braille à cause du nombre insuffisant de possibilités. Lors d'une lecture sur une plage braille, la correspondance n'est pas exacte avec l'écran puisque certains caractères en noirs sont transcrits en 2 symboles braille. Finalement, pour pouvoir représenter les 256 caractères du code ASCII, un point 7 sous le point 3 et un point 8 sous le point 6 ont été ajoutés. Désormais les plages braille possèdent donc des cellules à 8 picots, mais il est possible de désactiver les picots 7 et 8 pour une lecture plus facile.

Les handicapés visuels et Internet

Le développement du web et la généralisation d'Internet permettent maintenant à des personnes gravement handicapées visuelles d'accéder à une richesse d'informations et de services qui leur étaient précédemment inaccessibles. Contrairement à ce qui est souvent supposé, les personnes malvoyantes ainsi que les personnes aveugles peuvent accéder à l'informatique, et donc à Internet, en utilisant des aides techniques. Les lecteurs d'écran tels que JAWS[5] par exemple, sont utilisables avec la plupart des navigateurs web actuels (Internet Explorer, Netscape...). Pour surfer une personne aveugle peut également utiliser un navigateur vocal comme par exemple IBM Home Page Reader[6], éventuellement couplé avec son afficheur braille.

Sur internet, en complément de l'information que les personnes voyantes peuvent voir à l'écran, les lecteurs d'écran transmettent des informations qui proviennent du code source de la page web et qui sont invisibles aux personnes voyantes. Une page web ne possédant pas de focus comme pour un traitement de texte, le programme de lecture d'écran ne peut pas

---

[5] Téléchargement de la dernière version de JAWS en version limitée à 40 min d'utilisation : `http://www.freedomscientific.com/fs_downloads/jaws.asp` - dernier accès le 23/11/2006

[6] Téléchargement de l'ancienne version de IBM HPR utilisable pendant 1 mois gratuitement : `http://www-306.ibm.com/able/europe/fr/` - dernier accès le 23/11/2006

déterminer la position de lecture dans le texte. Pour naviguer à l'intérieur d'une page, des fonctions alternatives sont alors proposées à l'internaute. Par exemple l'utilisateur du navigateur vocal Home Page Reader peut utiliser les touches suivantes pour naviguer : « espace » pour commencer la lecture à partir de la position courante de la page ; « contrôle » pour arrêter la lecture ; « flèche gauche » et « flèche droite » pour lire le paragraphe respectivement précédent et suivant ; « tabulation » pour aller de lien en lien ; ou encore « entrée » pour activer un lien et ouvrir la page destination. Dès lors le texte de la page peut être transmis — à la synthèse vocale ou au clavier braille — ligne par ligne ou bloc par bloc. Précisons que de nombreux problèmes se présentent lorsque l'ordre de lecture d'une page est difficile à déterminer du fait de l'utilisation de tableaux ou de cadres (frames). Concernant les éléments non accessibles pour les lecteurs d'écran (comme les images de fond), il a longtemps été conseillé de préciser dans la feuille de style destinée aux personnes non-voyante qu'ils ne doivent pas être affichés (« display :none »). Or cela n'a pas uniquement pour conséquence de ne pas afficher l'image mais de supprimer tout le contenu. De ce fait au lieu d'avoir accès au texte de remplacement de l'image, le lecteur d'écran ne donnera aucune information à l'internaute. L'information est donc supprimée au lieu d'être rendu accessible. Remarquons également que la lecture d'éléments accessibles (comme les tableaux de données) peut également être difficile avec un lecteur d'écran lorsque le tableau est long. Lorsque l'internaute se trouve en milieu du tableau il doit effectivement se souvenir de l'en-tête des colonnes qu'il parcourt. Notons que des recherches ont été réalisées [101] pour faciliter la lecture des tableaux de données par les lecteurs d'écran. Dans ces travaux, chaque cellule devient un fichier HTML indépendant relié à ses cellules adjacentes ainsi qu'aux premières et dernières cellules de la ligne et de la colonne correspondante (contenant l'en-tête de la cellule). D'autre part, une étude sur la navigation des déficients visuels [131] a montré que ces personnes qui utilisent un lecteur d'écran passent en moyenne 6 fois plus de temps sur une page que les internautes voyants. La lecture auditive est donc plus lente que la lecture visuelle, et de nombreuses informations ne sont pas retranscrites à l'oral. Ce problème se pose quelle que soit l'application concernée sur Internet. Des travaux ont par exemple été menés pour créer un logiciel de messagerie vocal [92], proposant une assistance vocale adaptée aux besoins des personnes non-voyantes.

Pour faciliter la navigation et/ou modifier l'affichage des pages, des barres d'outils peuvent être associées aux principaux navigateurs. Tout d'abord, IE Accessible [202] est une barre d'outils pouvant être ajoutée à Internet Explorer pour faciliter la vision et donc la navigation dans un site web. Plus précisément, elle permet d'afficher la liste des liens et leur type (mail ou hyperlien), la liste des titres, la liste des raccourcis claviers de la page, la liste des frames et la liste des différents éléments des formulaires (bouton, label, edit. . .). Cette barre d'outils (figure 1.11) permet également d'utiliser sa propre feuille de style, et de désactiver les images, les couleurs, la police de caractère. Toutes les listes présentées ci-dessus sont non seulement accessibles *via* la barre d'outils mais aussi *via* des raccourcis clavier qu'il est possible de modifier dans le menu « propriété ». De plus, chaque liste peut être affichée par ordre alphabétique ou dans l'ordre d'apparition de la page. Accessibar [224] est une autre barre d'outils, pour Firefox et Mozilla (figure 1.12), pouvant fournir divers dispositifs permettant de modifier l'affichage comme la couleur du texte et du fond, la taille des caractères, l'interligne, ou encore de désactivation des images. De plus cette barre d'outils

intègre un synthétiseur vocal capable de lire l'interface utilisateur ainsi que le contenu de la page web. Néanmoins, bien que la barre d'outil Accessibar soit traduite en français, la voix du lecteur d'écran est anglaise ce qui rend difficile son utilisation sur des textes français. Une troisième barre d'outils est MozBraille [225] — nom obtenu par la contraction de Mozilla et Braille. C'est une extension pour transformer le navigateur Mozilla ou Firefox afin d'obtenir un navigateur internet autonome conçu pour les non-voyants et les malvoyants. MozBraille possède son propre lecteur d'écran et offre à l'utilisateur 3 types d'affichage : premièrement une sortie braille sur un terminal braille, deuxièmement une sortie vers un synthétiseur vocal, et troisièmement une vue avec des caractères agrandis.



FIG. 1.11 – Toolbar IE Accessible.



FIG. 1.12 – Toolbar Accessibar.

Outre les navigateurs classiques auxquels on peut ajouter des améliorations, il est possible d'utiliser un navigateur spécialisé comme par exemple Simply Web 2000 et BrailleSurf. Simply Web 2000 [186] est un navigateur basé sur Internet Explorer comprenant un synthétiseur vocal (figure 1.13). Il permet d'afficher la page telle qu'elle peut être vue par un navigateur classique et propose également une version accessible de cette même page. Il offre aussi la possibilité de visualiser sous forme d'une arborescence, la structure de la page comprenant des liens, des cadres et des formulaires. La dernière version de Simply Web 2000 est basée sur Internet Explorer 5.0 et fonctionne avec le système d'exploitation Windows, de Windows 95 à Windows 2000. BrailleSurf [205] est un navigateur destiné plus particulièrement aux handicapés visuels (figure 1.14). Il permet de traiter le code HTML de la page à afficher pour la transformer en texte. Les éléments graphiques sont supprimés et le texte peut être présenté à l'écran suivant les besoins des utilisateurs. Ce navigateur a été développé au sein de l'équipe INSERM « Interfaces Non Visuelles et Accessibilité » (Inova). Sa dernière version BrailleSurf 4 date de 2001 et marque l'arrêt des développements pour ce navigateur.

Malgré toutes ces aides techniques et logicielles, l'accès aux informations présentées sur le web peut continuer à poser des problèmes aux personnes aveugles ou malvoyantes. Sous leur forme actuelle, les renseignements du web ne sont effectivement pas facilement accessibles par les aveugles car peu de sites sont entièrement compatibles avec les aides techniques : un site web qui contient trop de cadres dans son code HTML peut empêcher une lecture adéquate à l'écran. Certains programmes en JavaScript nuisent également au bon fonctionnement des logiciels adaptés. De plus, de nombreux concepteurs de sites comptent sur des interfaces exclusivement graphiques pour la navigation sur leurs sites, oubliant ainsi d'offrir

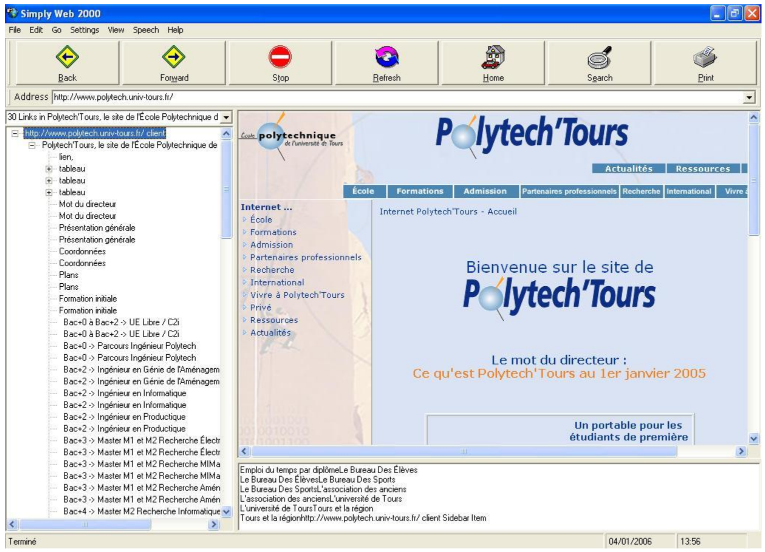


FIG. 1.13 – Simply Web 2000.

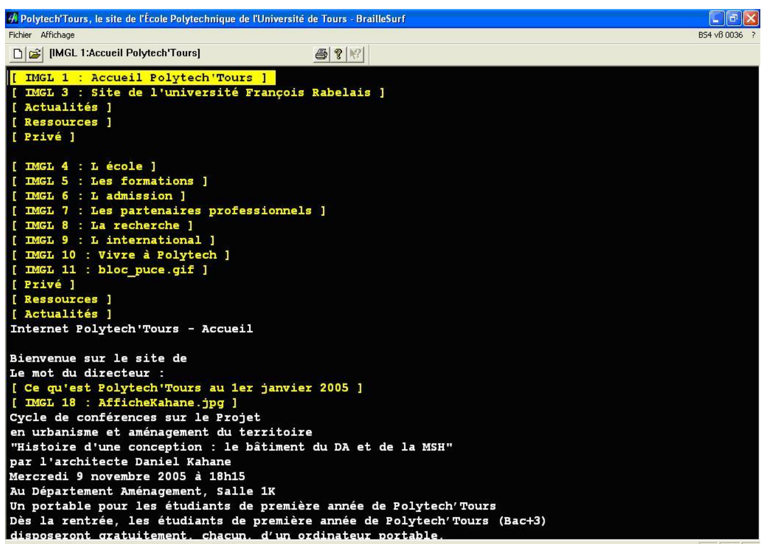


FIG. 1.14 – BrailleSurf.

un menu de base alternatif avec des informations textuelles. Des initiatives ont été prises afin d'adapter les sites web pour les personnes ayant certains handicaps. La WAI (Web Accessibility Initiative) est un groupe rattaché directement au W3C (World Wide Web Consortium) — ce dernier visant la standardisation des documents et des outils disponibles sur Internet. La WAI propose aux concepteurs de sites Web des lignes directrices qui les guideront vers la création de sites plus accessibles.

1.3 Cadre légal et organisationnel de l'accessibilité du web

Un site web est dit accessible s'il est consultable par tous (et en particulier par les personnes handicapées), quelque soit :
- la technologie de consultation utilisée (le site est consultable non seulement par les navigateurs web standard mais également par les aides techniques utilisées par les personnes handicapées),
- l'interface utilisée (clavier ou souris),
- la plate-forme d'accès (c'est à dire le système d'exploitation),
- le périphérique d'affichage (taille de l'écran, téléphone portable. . .).

1.3.1 Les organismes et associations travaillant sur l'accessibilité

Le W3C et la WAI

Le Consortium du World Wide Web (W3C) [253], fondé en octobre 1994 par Tim Berners-Lee (l'inventeur du World Wide Web), est un consortium industriel international. Il est simultanément géré par plusieurs organisations : le Groupement Européen de Recherche en Informatique et en Mathématiques (ERCIM) basé en France, l'Université de Keio au Japon, et le Laboratoire d'Informatique et d'Intelligence Artificielle du MIT (Massachusetts Institute of Technology) situé aux Etats-Unis. La principale mission du W3C est de créer des standards pour le Web en mettant à disposition des spécifications (appelées « recommandations » ou « directives »). Les objectifs du W3C [209] sont, entre autres, l'interopérabilité et l'accès universel qui sont définis ainsi :

> Interopérabilité :
> *« Il y a vingt ans, les gens achetaient des logiciels qui fonctionnaient uniquement avec les autres produits du même fournisseur. Aujourd'hui, une plus grande liberté de choix est présente, et les utilisateurs attendent des composants logiciels modernes qu'ils soient interchangeables. Ils souhaitent aussi accéder aux contenus Web avec leur matériel et logiciel préféré (navigateur de bureau graphique, synthétiseur vocal, affichage braille, téléphone mobile...). Le W3C est une organisation neutre qui favorise l'interopérabilité en concevant et promouvant des langages informatiques et des protocoles ouverts (non propriétaires). »*

> Accès universel :
> *« Le W3C définit le Web comme l'univers des informations accessibles en réseau. Un des premiers objectifs du W3C est de mettre ces avantages à la disposition de tous les individus, quelque soit leur matériel ou logiciel, leur infrastructure réseau, leur langue maternelle, leur culture, leur localisation géographique, ou leurs aptitudes physiques ou mentales. Les activités d'Internationalisation, de Web Mobile, de TV Web, de Navigateur Vocal, et l'Initiative d'Accessibilité au Web illustrent toutes notre engagement envers cet accès universel. »*

Pour travailler sur le domaine de l'accessibilité, le W3C a créé en avril 1997 la Web Accessibility Initiative (WAI) [258]. Cela concerne le développement de solutions pour rendre le Web accessible, principalement aux personnes handicapées. L'accessibilité du Web concerne principalement :

I. les sites Internet et les applications web
Dans ce sens, le document Web Content Accessibility Guidelines [27] (WCAG - Directives pour l'accessibilité aux contenus web) donne une définition des points à contrôler pour rendre un site accessible aux personnes handicapées. La version 1.0 de ce document fait référence, et a été produite le 5 mai 1999 par le groupe de travail sur les directives concernant le contenu web (WCAG **W**orking **G**roup). Le document WCAG 1.0 fournit les 14 directives suivantes :
1 - Fournir des alternatives équivalentes au contenu visuel et auditif
2 - Ne pas s'en remettre exclusivement aux couleurs
3 - Utiliser le balisage et les feuilles de style, et cela de façon appropriée
4 - Clarifier l'utilisation du langage naturel
5 - Créer des tableaux qui se transforment de façon élégante
6 - S'assurer que les pages contenant des technologies nouvelles se transforment de façon élégante
7 - Assurer à l'utilisateur le contrôle des changements du contenu lorsque ce dernier varie dans le temps
8 - Assurer un accès direct aux interfaces utilisateur intégrées
9 - Conception respectant l'indépendance par rapport au périphérique
10 - Utilisation de solutions intermédiaires
11 - Utilisation des technologies et des directives du W3C
12 - Fourniture d'informations de contexte et d'orientation
13 - Fourniture de mécanismes de navigation clairs
14 - S'assurer que les documents sont clairs et simples.

Ces directives permettent d'assurer une transformation d'un site non accessible vers un site accessible (directives 1 à 11), et de rendre le contenu compréhensible et navigable pour tous (directives 12 à 14). Chacune de ces directives possède un certain nombre de points à vérifier pour rendre un site accessible. Ceux-ci sont séparés en trois groupes de priorités différentes selon le niveau d'accessibilité à atteindre. Tout d'abord, les points de priorité 1 doivent être satisfaits pour rendre le contenu accessible à certains groupes. Ensuite, les points de priorité 2, s'ils sont satisfaits, lèvent certaines barrières et rendent par là même le document web davantage accessible. Finalement, si les points de priorité 3 sont satisfaits, seule une minorité de personnes éprouveront encore des difficultés à accéder aux informations. Ces 3 priorités permettent de définir quel niveau d'accessibilité est atteint par un site web. Si tous les points de priorité 1 sont satisfaits, le niveau A est attribué au site web. Ce dernier obtient le niveau double-A lorsque tous les points de priorité 1 et 2 sont respectés. Le niveau triple-A est alloué lorsque tous les points de priorité 1, 2 et 3 sont satisfaits. Pour signaler aux internautes qu'un site est accessible, la WAI a mis en place un système de label correspondant aux trois niveaux de conformités. Le webmestre qui a obtenu un des niveaux A, AA, ou AAA

peut alors apposer le logo correspondant (figure 1.15) sur son site web. Le WCAG liste au total 16 points de priorité 1, c'est à dire 16 points qui doivent impérativement être respectés pour obtenir le niveau de conformité simple A, 30 points de priorité 2 et 19 points de priorité 3. Le document « Techniques for Web Content Accessibility Guidelines 1.0 » [30] précise la manière de mettre en œuvre chacune des directives du WCAG 1.0. Ce document fait pour cela référence aux techniques HTML [29] et CSS [28].

Fig. 1.15 – Logos WAI des niveaux de conformité.

Une évolution des WCAG 1.0 est en cours : le document WCAG 2.0 [139] est actuellement candidat pour remplacer son prédécesseur. Le document « Techniques for Web Content Accessibility Guidelines 2.0 » [20] est en cours d'élaboration pour faciliter la mise en application des recommandations WCAG 2.0. Les WCAG 2.0 se composent de 13 directives regroupées en quatre principes, à savoir : (1) le contenu doit être perceptible, (2) les éléments d'interface dans le contenu doivent être utilisables, (3) le contenu et les commandes doivent être compréhensibles, (4) le contenu doit être assez robuste pour fonctionner avec les technologies actuelles et à venir. Les 13 directives proposées par le document WCAG 2.0 sont les suivantes :

1.1 - Fournir un équivalent textuel à tout contenu non textuel
1.2 - Fournir des alternatives synchronisées pour le multimédia
1.3 - S'assurer que les informations, les fonctionnalités et la structure sont séparables de la présentation
1.4 - Faire en sorte qu'il soit facile de distinguer l'information en premier plan des images ou des sons d'arrière-plan
2.1 - Rendre toutes les fonctionnalités utilisables *via* un clavier ou une interface clavier
2.2 - Permettre aux utilisateurs de contrôler les limites de temps sur leur lecture ou sur leur interaction
2.3 - Permettre aux utilisateurs d'éviter le contenu qui peut provoquer des crises épileptiques photosensibles
2.4 - Fournir des mécanismes pour aider les utilisateurs à trouver le contenu, à s'y orienter et à y naviguer
2.5 - Aider les utilisateurs à éviter les erreurs et leur en faciliter la correction
3.1 - S'assurer que le sens du contenu peut être déterminé
3.2 - Organiser le contenu de manière constante d'une page à l'autre et s'assurer que les composantes interactives fonctionnent de manière prévisible
4.1 - Utiliser les technologies conformément aux spécifications
4.2 - S'assurer que l'interface utilisateur est accessible ou qu'elle fournit une alternative accessible.

II. les navigateurs Internet et les lecteurs de média
Le groupe de travail sur les directives pour l'accessibilité des agents utilisateurs (UAWG)

a produit, le 17 décembre 2002, le document User Agent Accessibility Guideline [137] (UAAG 1.0 - Directives pour l'accessibilité des agents utilisateurs). Précisons qu'une évolution vers UAAG 2.0 [138] est en cours de réalisation. Les agents utilisateurs sont des logiciels permettant de restituer un contenu web, tels que les logiciels de consultation (navigateurs HTML classiques) mais aussi les aides techniques. Les 12 directives UAAG 1.0 permettent, grâce à des points de contrôle de priorités différentes, de concevoir des agents utilisateurs dont le but est de permettre une plus grande accessibilité au web pour les personnes ayant divers handicaps.

III. les logiciels de création de contenu multimédia
Le 3 février 2000, le document Authoring Tool Accessibility Guideline 1.0 [135] (ATAG 1.0 - Directives pour l'accessibilité aux outils de création de contenu) a été produit par le groupe de travail des règles d'accessibilité des outils d'édition (AUWG). Précisons que son successeur ATAG 2.0 [136] est en cours d'élaboration. Le document ATAG 1.0 est composé de 7 directives qui permettent aux développeurs de concevoir des outils de création de contenu qui soient accessibles aux auteurs handicapés, et qui génèrent du contenu accessible. Les logiciels de création de contenu comprennent les outils d'édition html et xml (souvent WYSIWYG pour les débutants), ceux permettant de sauvegarder un document dans un format web (par exemple certains éditeurs de texte), et les outils de gestion de sites (CMS, Content Managment System) qui génèrent des sites web dynamiquement à partir de bases de données.

Les projets dans les pays européens

En Belgique, le projet BlindSurfer [162] a été créé en 1999 dans le cadre d'un « projet d'accessibilité au web ». Il permet la diffusion d'informations concernant l'accessibilité *via* son site web et propose de former les responsables de sites web lors de séminaires, et d'évaluer l'accessibilité des sites web qui en ont fait la demande. Le label BlindSurfer offre à l'utilisateur d'un site web une indication quant à l'accessibilité de ce dernier. En effet, BlindSurfer propose une liste de directives relatives à l'accessibilité et classe les éléments d'évaluation par niveau de priorité : la priorité O attribuée aux éléments obligatoires, la priorité C affectée aux éléments que BlindSurfer conseille de respecter, et la priorité S accordée aux éléments supplémentaires à satisfaire pour optimiser l'accessibilité du site. Le label BlindSurfer est attribué par l'Œuvre Nationale des Aveugles (ONA, fondée en 1922), après un long travail d'évaluation du site par des experts agréés. Notons que BlindSurfer a changé de nom le 1er juillet 2006 et s'appelle désormais « AnySurfer » afin de montrer qu'un web accessible est bénéfique pour tous et pas seulement aux personnes aveugles. Le label AnySurfer possède deux niveaux d'accessibilité — AnySurfer de base et AnySurferPlus — contrairement à BlindSurfer qui en possédait trois.

Au Royaume-Unis, le projet See-it-Right [148] a pour objectif d'évaluer l'accessibilité des sites web. Pour cela les associations peuvent contacter le Royal National Institute for the Blind.

Aux Pays Bas, le projet Drempels Weg [147] permet lui aussi de tester les sites et de diffuser de l'information sur l'accessibilité.

En France, BrailleNet [174], et plus précisément sa composante Accessibilité (lancée en juin 2000), pilote le Groupe de Travail Accessiweb (GTA) depuis le 16 septembre 2003. Le but de ce groupe de travail est d'augmenter le nombre de professionnels du web maîtrisant les principes d'accessibilité, de proposer des méthodes et des outils pour rendre ou tester l'accessibilité des sites web. L'association BrailleNet a publié le guide Accessiweb composé de 92 critères et établi à partir des recommandations internationales WCAG 1.0 de la WAI. Il faut savoir que certaines directives restent inchangées tandis que d'autres ont été développées afin de lever certaines ambiguïtés. Le guide actuel correspond à la version 3.0 [171] et comporte 95 critères. Un tableau de correspondance Accessiweb/WCAG 1.0 [175] est disponible sur le site de BrailleNet. Le guide Accessiweb 1.0 a été intégré au « Référentiel accessibilité des services Internet de l'administration française » de l'ADAE (Agence pour le Developpement de l'Administration Electronique). L'ADAE a depuis été remplacée par la Direction Générale de la Modernisation de l'Etat [183] (DGME). La nouvelle version Accessiweb 1.1 remplace officiellement depuis juin 2008 la version Accessiweb 1.0. Les tables de correspondance entre le référentiel Accessiweb 1.1 et les référentiels WCAG et UWEM (voir ci-après) permettent ainsi d'évaluer le niveau d'accessibilité d'un site web à partir des normes Accessiweb, pour plusieurs référentiels. BrailleNet propose également le label Accessiweb [172] qui certifie l'accessibilité des sites web en appliquant les critères du guide Accessiweb. Ce label se décline en 3 niveaux, Bronze, Argent et Or, suivant le niveau d'accessibilité atteint.

La situation en Europe

Malgré les efforts fournis dans les pays européens, des études montrent que 90% des sites publics ne sont pas accessibles en Europe. Plus particulièrement, une étude menée en 2004 sur 123 sites en Irlande, 5702 en Angleterre, 4250 en Allemagne et 1545 en France [87] montre que respectivement 94%, 94,5%, 95,6%, et 98,6% des sites ne passent pas le test du logiciel Bobby (voir partie 1.4.3 page 42) avec le niveau minimum d'accessibilité, c'est-à-dire WCAG-A. De plus, cette étude montre que 99% des sites en Angleterre et 100% des autres sites testés ne sont pas conformes aux normes WCAG-AA. Malgré les lois mises en place dans certains pays d'Europe, les résultats d'accessibilité des sites ne sont pas concluants. Remarquons de plus que les lois étant différentes dans chaque pays européen, un site accessible dans son pays ne l'est pas forcément d'après la loi d'un autre pays européen.

Par conséquent, la mise en place d'une autorité européenne de certification de l'accessibilité des sites Web au niveau européen se révèle nécessaire. C'est dans ce but que 23 organismes européens, dont l'association Française Braillenet, participent à la création du Consortium EuroAccesssibility [191]. Ce Consortium, lancé le 28 avril 2003, permet également d'établir une méthodologie d'évaluation commune ainsi qu'une assistance technique. Pour ce faire, le projet support EAM (Supporting the creation of an eAccessibility Mark) [241] est lancé en octobre 2004 afin de créer un label européen sur l'accessibilité. Ce label devant être composé d'un ensemble de directives et d'une méthodologie de certification des sites web. Le projet Support EAM fait partie, avec deux autres projets européens, du WAB Cluster (Web Accessibility Benchmarking Cluster) [255]. L'objectif du WAB Cluster

est d'obtenir une compatibilité entre les outils d'évaluation et les méthodes de certification des différents pays européens. Pour cela le WAB Cluster a publié, le 5 juillet 2006 un document intitulé UWEM 1.0 (Unified Web Evaluation Methodology) [257][178]. Ce dernier établit une interprétation des WCAG 1.0, commune aux différents pays européens — c'est-à-dire comprenant les différentes directives de chaque pays, elles-mêmes des interprétations des WCAG 1.0. Dans la version UWEM 1.0, seuls les points de priorités 1 et 2 des WCAG 1.0 ont été définis. Actuellement la version officielle est le document UWEM 1.2 [256]. Le label Accessiweb est le premier label en Europe à appliquer le « schéma de migration vers une accessibilité du web harmonisée en Europe » [173]. Cela consiste principalement en la traduction des critères UWEM, et en la création d'un tableau de correspondance avec les critères UWEM. Le 14 septembre 2006, l'association Braillenet a publié son tableau de correspondance Accessiweb/UWEM 1.0 entre les critères Accessiweb 1.0 (niveau bronze et argent) et les critères UWEM 1.0 (priorités 1 et 2 des WCAG 1.0). Actuellement ce tableau est disponible pour la correspondance entre Accessiweb 1.1 et UWEM 1.0 [177]. Les critères UWEM étant une contraction des différents guides de chaque pays, lorsqu'un site web obtient par exemple le niveau argent du label Accessiweb, il possède de ce fait le niveau 2 UWEM. En revanche, un site web vérifiant le niveau 2 UWEM ne possèdera le niveau argent Accessiweb qu'après validation de critères supplémentaires — critères clairement identifiés par le groupe Accessiweb. Depuis septembre 2006, Accessiweb, Sello de Accessibilidad (Espagne) et, plus récemment (en mars 2007), AnySurfer (Belgique) ont rejoint ce projet européen. EuraCert [190], une autorité de certification européenne a récemment (en juin 2007) été mise en place. Ainsi, un site web portant le label EuraCert est reconnu comme étant accessible par tous les pays faisant partie de ce projet (donc à ce jour la France, l'Espagne et la Belgique). Le site web de Gaz de France[7] est le premier site labélisé EuraCert en septembre 2007. Dès que les autres pays européens auront appliqué le « schéma de migration », les labels nationaux existant pourront migrer vers le schéma d'accessibilité européen.

Si on résume, chaque pays a développé ses propres normes d'accessibilité et la convergence vers une uniformité des normes en Europe est lente. Mais au final très peu de sites web sont accessibles, quelque soit la norme adoptée.

1.3.2 La réglementation concernant l'accessibilité

L'accessibilité est désormais une obligation pour les sites institutionnels dans de nombreux pays. Les recommandations de la WAI servent de référence, et ont souvent été explicitement intégrées dans des cadres juridiques.

Aux Etats-Unis

Au Etats-Unis, en août 1998, la législation imposait l'accessibilité aux sites Web fédéraux depuis la modification apportée à la section 508 [236] de la loi sur la réadaptation des personnes handicapées (« Rehabilitation Act »). En décembre 2000, le United States Access

[7] `http://www.gazdefrance.com/FR/`

Board a élaboré des règles relatives à la mise en application de la section 508. Ces règles ont pris effet en juin 2001. Elles précisent que le niveau A doit être atteint ainsi que 5 autres règles de niveau AA et AAA.

Dans certains pays européens

En Belgique, le Gouvernement a approuvé une note d'orientation dans laquelle figurent les mesures relatives à l'accessibilité des sites web (loi anti-discrimination du 25 février 2003 [154], supprimée par la loi du 10 mai 2007 [156]).

En Grande-Bretagne, la loi D.D.A. (Disability Discrimination Act) [153] de 1995 demande non seulement aux sites du secteur public, mais aussi aux sites commerciaux, d'être accessibles. Cette loi n'étant pas suivie par un décret d'application, le cadre légal fut complété par le projet See-it-Right. Ce n'est qu'en octobre 2004 que le « code of practice » est entré en vigueur précisant le niveau d'accessibilité attendu.

En Allemagne, la loi du 1er mai 2002 sur l'intégration des personnes handicapées stipule que les sites web des organismes du secteur public doivent être accessibles (satisfaire aux priorités 1 et 2 des directives WCAG du W3C) avant le 31 décembre 2005, et que les sites public spécifiquement orientés vers les personnes handicapées doivent l'être avant le 31 décembre 2003.

En France, la loi du 30 juin 1975 sur *la politique en faveur des personnes handicapées* [2], constitue un ensemble de dispositions légales permettant de poser les bases de l'amélioration de la situation des personnes handicapées en France. De plus, quelques mois plus tard (le 9 décembre 1975), l'Assemblée générale de l'ONU proclame la *Déclaration des droits des personnes handicapées* [6] dans laquelle l'article 5 stipule que « la personne handicapée a droit aux mesures destinées à lui permettre d'acquérir la plus large autonomie possible ». Avec l'émergence d'Internet, une réforme de la loi du 30 juin 1975 est devenue nécessaire pour préciser quelques points quant aux nouvelles technologies. Ainsi, le *Projet de loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées* [7], traite dans les articles 21 à 25 de l'accessibilité aux bâtiments, aux transports et aux nouvelles technologies. L'article 25 (ci-dessous) définit un nouveau principe d'accessibilité des services de communication en ligne aux personnes handicapées, pour notamment faciliter leurs démarches administratives. Cela signifie que la loi demande aux sites publics (Etat, collectivités territoriales, établissements publics) de suivre les recommandations internationales liées à l'accessibilité. Ce projet de loi a été adopté le 3 février 2005 et la loi est parue au Journal Officiel le 12 février 2005 [3]. L'article 25 du projet de loi, est devenu l'article 47 de la loi *pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées*. Cet article renvoie pour la mise en œuvre de ces dispositions de principe, à un décret en Conseil d'État qui devra intervenir pour préciser le contenu et les délais des adaptations nécessaires. D'autres dispositions sont également prévues (mise en conformité d'ici à 3 ans, formation des personnes impliquées, prise en compte des handicaps visuels, mais aussi auditifs ou moteurs) [181][184]. Malheureusement, 3 ans après le vote de cette loi, aucun décret d'application n'a été publié. Pour combler ce manque, l'association BrailleNet propose en décembre 2007 aux autorités compétentes un projet de décret [179].

Remarquons que récemment, deux documents officiels [121][221] ont annoncé la publication du décret d'application pour juillet 2008.

> Article 25 du Projet de loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées :
>
> *Les services de communication publique en ligne des services de l'Etat, des collectivités territoriales et des établissements publics qui en dépendent doivent être accessibles aux personnes handicapées.*
>
> *L'accessibilité des services de communication publique en ligne concerne l'accès à tout type d'information sous forme numérique quels que soient le moyen d'accès, les contenus et le mode de consultation. Les recommandations internationales pour l'accessibilité de l'Internet doivent être appliquées pour les services de communication publique en ligne.*
>
> *Un décret en Conseil d'Etat fixe les règles relatives à l'accessibilité et précise, par référence aux recommandations établies par l'Agence pour le développement de l'administration électronique, la nature des adaptations à mettre en œuvre ainsi que les délais de mise en conformité des sites existants et les sanctions imposées en cas de non-respect de cette mise en accessibilité. Le décret énonce en outre les modalités de formation des personnels intervenant sur les services de communication publique en ligne.*

Pour toute l'Europe

En décembre 1999, l'Union européenne se dote d'un plan d'action eEurope [194], dont le but est de faire bénéficier l'ensemble des citoyens européens des avantages offerts par la société de l'information. Le premier Plan d'action, eEurope 2002, visait principalement à accroître l'usage d'Internet, notamment dans les domaines d'intérêt public. Le but était de rendre le web accessible à tous et de favoriser le développement du commerce électronique. Ainsi le plan eEurope 2002 obligeait les sites web publics des Etats membres de l'union européenne à être accessibles aux handicapés. Le 28 mai 2002, la Commission Européenne accepte le plan d'action eEurope 2005 [192], qui confirme l'adoption des recommandations W3C/WAI pour l'accessibilité des sites web. Cela montre la volonté des États membres et des institutions européennes de parvenir à intégrer l'accessibilité du Web dans les politiques nationales et institutionnelles. Depuis, le plan d'action eEurope 2010 [193] a été adopté. Cette initiative i2010 précise le besoin d'augmenter l'intensité de la recherche et l'investissement dans les technologies de l'information et de la communication.

Au sein de eEurope, qui s'occupe globalement du développement dans le domaine de la technologie d'information, nous pouvons observer différents groupes de travail comme « eAc-

cessibility » qui s'occupe plus particulièrement de l'accessibilité électronique (donc également de l'accessibilité du web), et « Design of all », dont le thème de recherche principal est la création d'une technologie d'information accessible (avec comme thème central l'accessibilité au web). Les résultats de ces groupes de travail conduisent ensuite à l'établissement d'une réglementation dans l'union Européenne.

Le site Web EUROPA [194] permet d'obtenir des informations concernant les activités des institutions européennes. Rendre ce site accessible était une priorité, d'autant plus depuis l'apparition, en juin 2001, d'un guide des fournisseurs d'information qui explique en 10 règles comment créer du contenu web, et notamment dans la règle 7 : comment faire pour qu'il soit accessible en respectant les recommandations de la WAI. Ce guide est actuellement remplacé par le « Information Providers Guide » [246] qui donne la position de l'Europe en matière d'accessibilité ainsi que des liens vers les principales recommandations.

1.4 En pratique : créer ou rendre un site web accessible

1.4.1 Accessibilité et utilisabilité

Relation entre accessibilité et utilisabilité

Une étude menée en 2000 par T. Sullivan et R. Matson [127] sur 50 sites très populaires a permis de montrer que 41 d'entre eux n'étaient pas accessibles, et qu'il y avait une forte corrélation entre accessibilité et utilisabilité. L'accessibilité est effectivement un sous concept de « l'utilisabilité » (traduction littérale du terme « usability ») [125]. Communément, le terme « utilisabilité » est souvent utilisé pour désigner la capacité d'un produit à être utilisé facilement. Plus précisément, d'après l'ISO [69], l'utilisabilité représente *le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié.* L'utilisabilité dépend finalement des objectifs, du contexte, et de l'utilisateur. Un logiciel possédant un bon degré d'utilisabilité pour une personne valide, ne sera pas forcément aussi utilisable pour une personne handicapée. Pour mesurer cette utilisabilité, trois points sont à vérifier : (1) l'efficacité (le but a-t-il été atteint ?) ; (2) l'efficience (le but a-t-il été atteint avec un minimum d'efforts, en un minimum de temps ?) ; (3) la satisfaction (quel a été le ressenti global ?). En définitive, un site accessible doit être ergonomique, c'est-à-dire utilisable/convivial, à savoir facile à prendre en main, simple et agréable à utiliser, et utile c'est-à-dire capable de répondre aux attentes et besoins réels des internautes à travers un contenu pertinent et de qualité.

D'autres recherches ont été effectuées en 2001 [142] [143] [141] en s'inspirant de l'étude de T. Sullivan et R. Matson. Elles montrent que la catégorie de sites les plus accessibles sont les sites gouvernementaux (52% de ceux-ci sont accessibles). Ils sont suivis par les sites de l'éducation (25% sont accessibles). De plus, ces derniers, ainsi que les sites gouvernementaux et organisationnels, montrent une forte corrélation entre l'accessibilité et l'utilisabilité, contrairement aux sites commerciaux. Ces derniers sont beaucoup utilisés mais sont incommodes pour les personnes handicapées.

L'accessibilité est un sous concept de l'utilisabilité, mais un haut niveau d'accessibilité n'engendre pas forcément un haut niveau d'utilisabilité. Il est possible de créer des sites accessibles mais difficilement utilisables, et les outils de validation d'accessibilité ne vérifient pas l'utilisabilité. C'est pour cela qu'il est conseillé de croiser des évaluations d'ergonomie (par exemple avec l'outil Destine [85]) et d'accessibilité, couplées avec des tests d'usages [11].

Règles générales d'utilisabilité des sites web

Il y a certaines règles d'ergonomie à respecter pour que l'information présente sur le site soit compréhensible et utilisable simplement. Les critères ergonomiques proviennent d'études empiriques ou de pratiques courantes. La version actuelle a été publiée par Scapin et Bastien [118]. Ces critères se sont avérés utiles pour la classification de plus de trois cents recommandations ergonomiques pour la conception de sites web [80]. Ces dernières

sont répertoriées en catégories parmi lesquelles on trouve le guidage (incitation, lisibilité...) et l'homogénéité/cohérence. Voici quelques recommandations importantes, concernant ces deux catégories, qui relèvent bien souvent du bon sens.

Tout d'abord le langage utilisé doit être adapté aux propos contenus dans le site web mais également au public visé. Il en est de même pour le mode de navigation qui pourra être plus « fantaisiste » si le site est destiné à un public jeune, alors qu'il devra être plus conventionnel s'il s'adresse à un public de seniors. Une première question à laquelle il faudra répondre pour évaluer un site est donc le public visé.

Ensuite il faut vérifier que la présentation est homogène sur tout le site, ce qui facilite l'identification des différents éléments. Le visuel est fort utile pour structurer les propos et agrémenter la mise en page. Il est essentiel de veiller à créer tout au long des pages d'un site, une identité et/ou une cohérence visuelle qui guidera l'internaute tout au long de sa visite. Le « visuel » — par lequel on entend également la position des éléments — est également nécessaire pour les malvoyants ou non-voyants qui trouveront sur chaque page les éléments qu'ils recherchent toujours au même endroit.

Il est bon ensuite de vérifier que certaines conventions sont bien adoptées selon le type de site. Par exemple, dans les sites institutionnels on trouvera le plus souvent le logo en haut à gauche, le menu en haut ou sur la gauche, et sur la droite un moteur de recherche, des liens vers les partenaires et éventuellement de la publicité.

De plus, pour avoir un bon degré d'utilisabilité, un site web doit posséder une structure logique et équilibrée. Pour cela il faut vérifier que le site est lisible, c'est-à-dire que le contenu de chaque page est bien structuré : de petits paragraphes avec des titres bien différenciés faciliteront la lecture. Pour rendre la navigation plus aisée pour tout internaute — et donc à fortiori pour les malvoyants et les non-voyants —, il convient de construire un site bien structuré. Par structure apparente d'un site web, nous entendons ici la structure du site telle qu'elle apparaît a l'internaute — par opposition à la structure en tant qu'organisation des fichiers composant le site. La structure apparente facilite le déplacement du visiteur sur un site pour lui permettre de trouver rapidement l'information qu'il cherche. Il faut structurer le site de telle manière qu'il y ait un équilibre entre les différents blocs d'informations. Pour les sites qui connaissent une croissance rapide et de nombreux développements, l'objectif sera de garder cet équilibre. Une représentation graphique de la structure d'un site permet de visualiser rapidement les éventuels déséquilibres. Pour les sites contenant de nombreuses pages, il existe des outils permettant d'extraire automatiquement la structure du site. Une fois la structure bien définie il est bon d'en faire profiter les internautes en leur proposant une page du site web dédiée à l'affichage du plan. La section 2.2 page 55 détaille les différents plans de sites existant.

D'autres règles générales sont parfois mises en évidence [169], comme réduire le temps de chargement de la page ou proposer un moteur de recherche performant. Concernant l'utilisation des cadres (frames) ou du flash, la frontière entre accessibilité et utilisabilité est insignifiante. Nous tenterons par la suite de revenir autant que possible sur l'accessibilité du web en découvrant les règles pour réaliser un site web accessible.

1.4.2 Les guides : synthèse sur l'accessibilité des sites web

Le respect de la législation en vigueur concernant l'accessibilité — ou l'amélioration de son site web en matière d'accessibilité — nécessite de modifier le code source des pages afin qu'il soit conforme aux normes. Toutefois la migration vers un site accessible peut s'avérer complexe surtout si ledit site a été mis en place par un webmestre amateur. Des guides en français sont proposés — comme ceux de Voir+ [250], BrailleNet [176], La Grange [212], et Accès-pour-tous [160] — pour informer et aider quiconque souhaitant rendre ou créer un site accessible. Ils établissent des liens vers les principaux outils de validation automatique (voir section 1.4.3). Le guide Voir+ (figure 1.16 (c)) est constitué de liens vers la documentation officielle, c'est-à-dire les recommandations HTML, CSS et WCAG 1.0. Certains, comme BrailleNet, La Grange et Accès-pour-tous (figure 1.16, respectivement (b), (a) et (d))proposent également un ensemble de fiches techniques permettant la résolution des erreurs mises en évidence par les outils de validation. Accès-pour-tous propose en supplément des outils de validation, grâce auxquels il est possible de tester le site par niveaux de priorités à atteindre. Le code source de la page est affiché et les lignes contenant des erreurs sont mises en évidence. Pour chacune des erreurs, on connaît son niveau de gravité et une aide à la résolution est proposée. Au-delà d'une approche purement informatique, la particularité du guide « Plongez dans l'accessibilité » de La Grange est de faire connaître certains handicaps. Pour cela il met en scène des personnages dans le but de sensibiliser le lecteur sur les difficultés qu'ils peuvent rencontrer sur Internet. Ce guide propose de rendre un site web accessible en trente jours et prend le temps (pendant cinq jours) de mettre l'utilisateur en situation en présentant les problèmes rencontrés par cinq personnes handicapées.

Tous ces guides établissent des liens vers les recommandations du W3C et vers les outils de validation automatique. Certains expliquent même le contexte législatif actuel en ce qui concerne l'accessibilité des sites web. Le tableau 1.4 résume les différentes fonctionnalités de ces guides.

1.4.3 Les validateurs : vérification automatique

Durant la création d'un site ou lors de sa modification dans le but d'obtenir un site accessible, il faut vérifier la conformité du code par rapport aux normes faisant références dans le domaine de l'accessibilité. Pour cela, il existe plusieurs outils de vérification automatique du code disponibles sur Internet. Précisons qu'avant de vérifier les normes d'accessibilité, il est important de vérifier l'adéquation du code de la page avec les normes des langages de description.

Tester le respect des normes HTML

HTML Validator [252], mis en ligne par le W3C, contrôle le code source HTML en vérifiant s'il respecte les normes HTML 4.01 [107]. Ce même outil permet également de tester la validité du code si celui-ci est écrit en XHTML. Pour tester le code, il suffit de rentrer l'adresse URL de la page ou le chemin du code présent en local. Quelques secondes

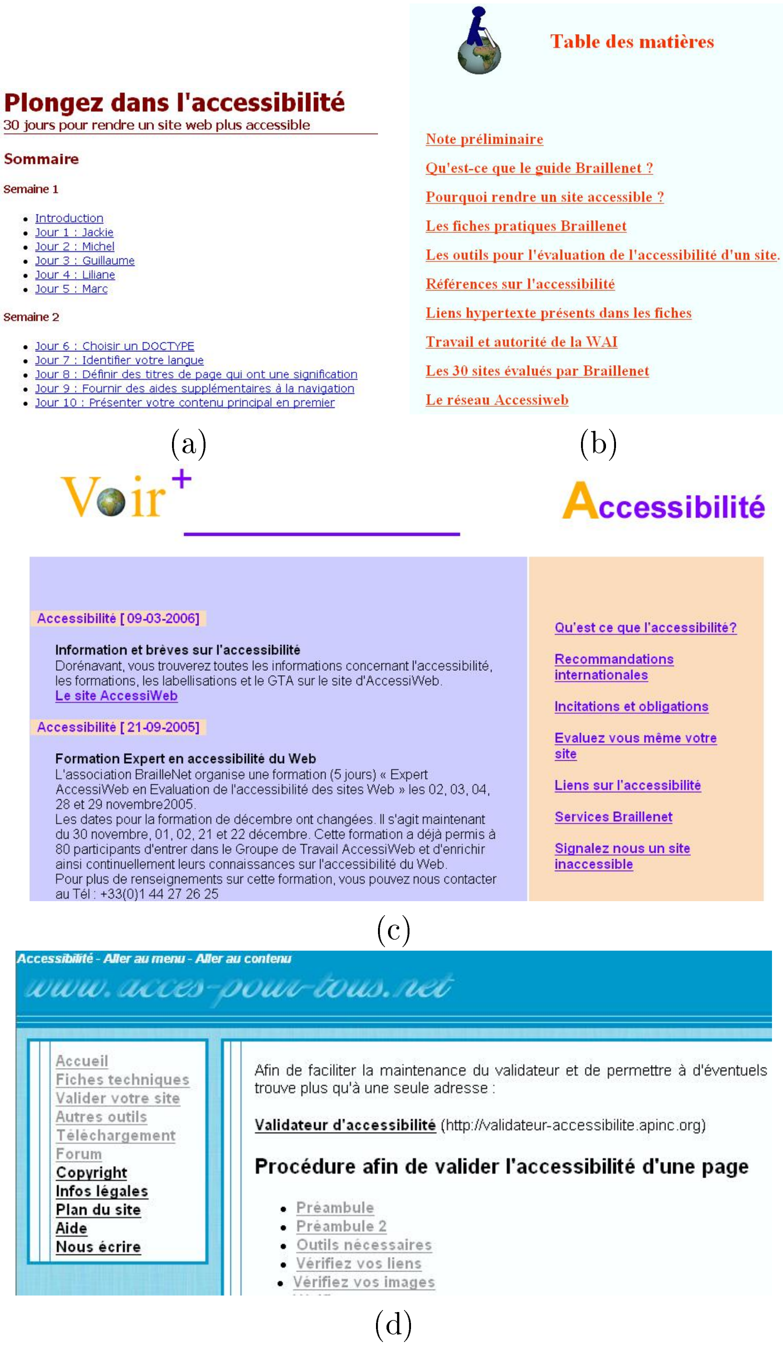


FIG. 1.16 – Les 4 principaux guides francophones sur l'accessibilité du web : La Grange (a), BrailleNet (b), Voir+ (c) et Accès-pour-tous (d).

	La grange	Voir+	BrailleNet	Accès pour tous
Informations sur le handicap et les aides techniques	x		x	
Contexte législatif		non actualisé	non actualisé	
Lien vers les recommandations officielles (HTML, CSS, WCAG)	x	x	x	x
Explication des recommandations officielles			x	
Lien vers des outils de validation automatique	x	x	x	x
Fiches techniques	x		x	x
Validateur intégré				x
Démarche d'accessibilité				générale

TAB. 1.4 – Comparaison de différents guides d'accessibilité.

plus tard, les résultats s'affichent. Si la page n'est pas valide, le nombre d'erreurs est indiqué et un listing apparaît avec des explications et des liens vers des pages d'aide à la résolution de ces problèmes.

D'autres validateurs sont également disponibles comme le validateur du W3Quebec [254], le validateur du Web Design Group (WDG) [260], les validateurs francophones Validome [248] et celui du W3C écrit en PHP [223], ou encore le validateur ProzeValide [234] (aussi appelé Validateur W3C Multipages).

Tester le respect des normes CSS

CSS Validator [251] permet de tester les feuilles de style (CSS - Cascading Style Sheet). Pour réaliser ce test, il suffit d'entrer l'URL de la feuille de style ou le chemin de la CSS en local. Comme pour HTML Validator, la liste des erreurs apparaît ensuite. Suite à ce listing d'erreurs, le CSS Validator propose une version de la feuille de style CSS validée dans laquelle les lignes de code invalides ont simplement été supprimées. Cette démarche n'est pas la bonne car certains styles non conformes aux normes CSS mais utilisés sur le site web sont supprimés par le validateur, ce qui va modifier l'affichage des pages du site web. Il est alors conseillé de résoudre les erreurs une à une et de chercher une alternative pour chaque cas en s'aidant des recommandations CSS du W3C [140].

Tester l'accessibilité

Bobby [259] est un validateur en ligne, des normes WCAG 1.0 et produisant, pour une page donnée, un rapport des erreurs qu'il divise en 4 sections :

- erreur de priorité 1 : problèmes qui affectent sérieusement l'utilisation de la page
- erreur de priorité 2 : problème d'accès secondaire
- erreur de priorité 3 : problème d'accès tertiaire
- erreur de compatibilité avec le navigateur : les éléments HTML utilisés sur la page ne sont pas valides pour les navigateurs particuliers.

Le rapport d'analyse reprend toutes les erreurs ligne par ligne en fournissant une explication pour pallier au problème. Une fois la page valide, Bobby fournit un logo correspondant au niveau d'accessibilité de la page ainsi qu'une liste de critères plus subjectifs qu'il est difficile de tester automatiquement.

OCAWA [247], à l'instar de Bobby, est un test d'accessibilité en ligne. Cependant il distance les autres outils, notamment par l'intégration de tous les standards et règles existants mais aussi par la mise en valeur des aspects d'inaccessibilité directement dans le code source et précisément sur la balise concernée dans le rapport. De plus, l'ensemble du code source est affiché précisant l'emplacement précis des erreurs détectées dans le rapport. Toutefois étant un outil commercial, le nombre de pages qu'il est possible de tester gratuitement est limité.

A-Prompt [165] est un logiciel téléchargeable qui permet de vérifier des pages hors-ligne. Le rapport obtenu liste l'ensemble des erreurs à modifier et propose une correction automatique. Après avoir entré la localisation de la page HTML, la fenêtre de correction apparaît. En sélectionnant l'erreur et en cliquant sur « Réparer » l'erreur peut être automatiquement rectifiée. Un certain nombre d'erreurs demande tout de même une correction manuelle. A chaque erreur est associée sa priorité par rapport aux directives de la WAI. De plus, l'utilisateur a la possibilité de choisir le niveau d'accessibilité souhaité. Ainsi, le nombre d'erreurs détectées varie en fonction du niveau choisi. Une fois la page réparée, le logo A-PROMPT peut y être inséré avec le niveau d'accessibilité correspondant.

The Wave [261] est un navigateur spécialisé qui permet de tester un grand nombre d'aspects d'accessibilité de la WAI, par exemple la présence d'attributs `alt` dans certaines balises, l'ordre de linéarisation des tableaux, l'emploi d'attributs HTML obsolètes ou erronés, la présence de Javascripts... L'emploi de The Wave est particulièrement aisé : un clic suffit pour que la page web s'affiche avec des illustrations supplémentaires aux endroits sensibles pour coder les erreurs, les alertes et les différents éléments du rapport. Il est possible d'activer les liens et de naviguer de page en page. Ce service est disponible en ligne, pour des pages résidentes ou non. La figure 1.17 représente l'affichage par The Wave de la page d'accueil du site web de l'équipe Handicap et Nouvelles Technologie (HaNT) du Laboratoire d'Informatique (LI) de l'université de Tours. On peut y voir en vert les éléments respectant les normes d'accessibilité, en orange les avertissements, et en rouge les erreurs. Un autre avantage de The Wave est l'affichage de l'ordre de déplacement par la touche tabulation. Cela permet de visualiser l'ordre d'affichage des éléments en mode texte, et de suivre le déplacement d'un internaute n'utilisant pas de souris.

Le tableau 1.5 résume les caractéristiques essentielles des quatres principaux validateurs cités ci-dessus.

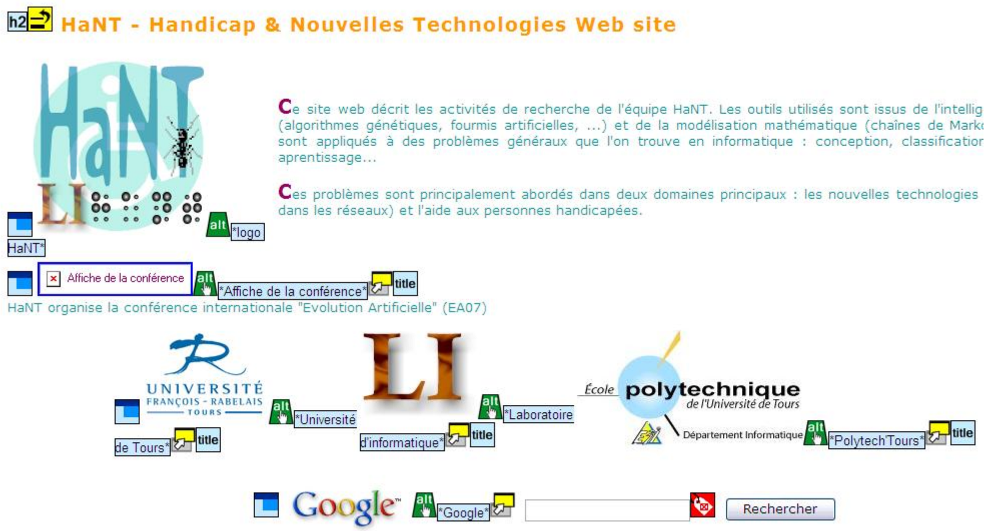


FIG. 1.17 – Visualisation du site web de l'équipe HaNT par The Wave.

	Bobby	Ocawa	A-Prompt	The Wave
outil	en ligne	en ligne	hors ligne	en ligne
test de pages	en ligne	en ligne	hors ligne	en/hors ligne
langue	angl	angl/fr	angl	angl
choix du niveau d'accessibilité	non	non	oui	oui
priorité des erreurs	oui	non	oui	non
listing des erreurs par thème	non	oui	non	oui
numéro de la ligne de l'erreur	oui	oui	non	non
visualisation	non	code	non	page
explication détaillée des erreurs	oui	non	non	non
proposition de correction	non	non	oui	non

TAB. 1.5 – Comparaison de différents validateurs d'accessibilité.

Hormis ces quatre validateurs, on peut en trouver de nombreux autres comme par exemple A-Checker [164], ART Guide [245], EvalAccess [213], Hermish [239], Web Accessibility Inspector [196], Tidy [235], Cynthia Says [201] ou encore Hera [197], qui sont des validateurs gratuits en ligne.

1.4.4 Les outils facilitant la vérification manuelle

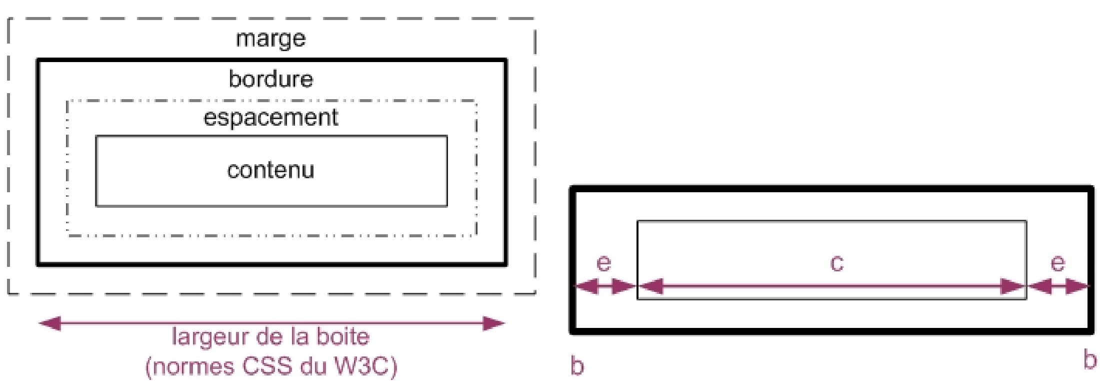
Les outils automatiques permettent de corriger beaucoup d'erreurs mais certaines normes d'accessibilité sont plus subjectives et il n'est pas possible de les corriger par un outil automatique. Par exemple le fait que toutes les images soient dotées d'un texte alternatif est

vérifiable automatiquement tandis que la pertinence du texte alternatif ne l'est pas. Il est donc primordial de connaître les directives de la WAI données dans le document WCAG 1.0 et les divers points à contrôler. Pour contrôler que le code source est bien en adéquation avec les recommandations, il est possible de parcourir tout le code source et de vérifier une à une les recommandations, ou bien de s'aider de divers outils (navigateurs, outils spécialisés) qui facilitent l'analyse manuelle.

Observation avec différents navigateurs

Un site web accessible se doit d'être consultable quelque soit le navigateur utilisé et le périphérique d'affichage. Il convient de visualiser le site avec différents navigateurs classiques comme Internet Explorer, Netscape, Mozilla et Opera, qui sont les plus utilisés. Même si le site est construit à l'aide de feuilles de styles respectant les normes CSS et du code valide HTML ou XHTML, il est probable en effet que l'affichage diffère selon le navigateur web utilisé. Internet Explorer 6.0 qui est sorti en 2001 et ne fut pas remplacé avant 2006, ne respecte pas totalement les normes CSS.

Prenons par exemple la différence entre l'évaluation des dimensions des boites (c'est-à-dire une aire de contenu) selon les normes CSS du W3C et selon Microsoft. Une boite possède une aire de contenu entourée par une aire d'espacement, une aire de bordure (correspondant à l'épaisseur de la bordure) et une aire de marge (figure 1.18 (a)). D'une part, d'après le W3C, la taille globale d'une boite se calcule en additionnant la largeur du contenu, avec les marges gauche et droite et avec l'épaisseur des bordures de la boite. D'autre part, Microsoft et donc Internet Explorer a sa propre interprétation de la largeur d'une boite : l'épaisseur des bordures et les marges font partie du contenu. Ainsi pour Internet Explorer, la largeur totale d'une boite est égale à la largeur de son contenu. Sur la figure 1.18 (b), le webmestre a choisi d'avoir un contenu de largeur c, des espacements à gauche et à droite de largeur e et une bordure d'épaisseur b. La largeur de sa boite sur la page web vaut $b + e + c + e + b$ selon le modèle du W3C, alors qu'elle vaut c selon le modèle Microsoft (les dimensions b et e venant réduire la taille de la zone de contenu utilisable).



(a) (b)

FIG. 1.18 – Les différentes aires d'une boite (a), et un exemple d'application (b).

Cependant il est possible de forcer l'une ou l'autre des interprétations des dimensions grâce à la technique du Doc Type Switching [60], c'est-à-dire en fonction de la Déclaration du Type de Document (DTD) spécifiée en tête de la page à afficher. En l'absence de doctype ou avec le doctype suivant : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">, c'est le modèle Microsoft qui sera adopté. Le choix d'un doctype est très important car même si les navigateurs Mozilla et Opera sont aujourd'hui les meilleurs en terme de qualité de leur interprétation des standards du web, Internet Explorer reste néanmoins le plus utilisé. Internet Explorer 7 (IE7), sorti en 2006, possède une comptabilité avec les normes CSS qui s'est grandement améliorée depuis son prédécesseur. Cependant IE7 reste sur ce point fortement inférieur à Firefox [262].

Les navigateurs Internet Explorer, Netscape et Mozilla sont gratuits et téléchargeables sur Internet. Opera est l'un des rares navigateurs payants. Reste que sa version gratuite contenant un bandeau publicitaire, comporte toutes les fonctionnalités de la version payante. L'accessibilité du site pourra alors être vérifiée en utilisant la version gratuite d'Opera.

Hormis les navigateurs graphiques classiques, il existe des navigateurs en mode texte comme Lynx (ou Lynx Viewer[8], la version pour Windows) et Braillesurf. Tous deux permettent de vérifier que l'organisation de la page, c'est-à-dire l'ordre des éléments, reste inchangée par rapport à celle observée avec un navigateur graphique.

Observation avec différentes résolutions

Afin de vérifier que le site web s'affiche correctement quel que soit le dispositif d'affichage, il faut tester les pages avec différentes résolutions. Le site web doit être consultable sur de petits écrans comme sur des grands, par des résolutions plus ou moins élevées. Pour tester ce critère il est possible de visualiser le site avec des résolutions d'écran différentes. Un site optimisé pour toutes les résolutions doit s'afficher de la même manière que ce soit avec une faible résolution (800*600 pixels) ou bien avec une haute résolution (1600*1200 pixels ou plus) en passant par des résolutions courantes (1024*768 ou 1152*864 pixels), et ne pas nécessiter l'utilisation d'un ascenseur horizontal pour restituer le contenu.

Test avec des outils d'aide à l'accessibilité

Un site accessible doit être consultable par tous, donc également par les personnes utilisant des aides techniques. Précisons que les aides techniques telles que les claviers braille et les synthétiseurs vocaux sont très coûteux. De ce fait, il peut être difficile de tester un site par ce biais. Toutefois, les navigateurs actuels proposent tous des options d'accessibilité, qui peuvent servir aux malvoyants pour remodeler les pages selon leurs préférences et leurs besoins, mais aussi aux développeurs pour vérifier l'accessibilité de leur site :

1. l'utilisation d'un zoom
   Les malvoyants peuvent avoir besoin de zoomer les pages d'un site. Internet Explorer

[8] http ://www.delorie.com/web/lynxview.html

permet de modifier la taille des caractères dans le menu *Affichage*, puis *Taille du texte*. Les différentes tailles de caractères doivent être testées et la police choisie sur le site doit permettre un affichage clair quelque soit la taille des caractères. Cet aspect peut également être testé avec le navigateur Netscape par le menu *Afficher*, puis *Zoom Texte* et par le navigateur Mozilla de la même manière que sous IE. Opera propose quant à lui un véritable zoom (menu *Affichage*, puis *Zoom*) qui permet de grossir ou rétrécir toute la page, ce qui inclut le texte mais également les images.

2. la désactivation des feuilles de styles
Les normes de la WAI précisent dans les directives des WCAG 1.0 l'importance de séparer le fond et la forme. Pour cela, il est recommandé d'utiliser des feuilles de style pour paramétrer l'affichage, la disposition des éléments, les polices de caractère, indépendamment du contenu. Cette séparation permet aux malvoyants de désactiver les feuilles de style qui ne leur conviennent pas et/ou d'appliquer aux sites leur propre feuille de style qui sera mieux adaptée à leurs besoins. Internet Explorer permet de tester les sites sans leur feuille de style ou en la remplaçant par une autre grâce au menu accessibilité des options d'Internet. Le navigateur Opera permet également de désactiver les feuilles de styles en passant en mode *Auteur*. Le navigateur en mode texte Lynx ne tient pas compte des feuilles de styles. Ainsi il est pratique de regarder l'aspect du site grâce à Lynx Viewer pour voir si la page reste compréhensible sans sa feuille de style.

3. la modification des couleurs pour améliorer le contraste
Les personnes malvoyantes doivent avoir la possibilité d'augmenter le contraste comme bon leur semble. Le webmestre doit pour cela s'assurer que cette modification n'affecte pas le contenu du site. Internet explorer permet de modifier les couleurs dans la partie *Couleurs* des options d'Internet ; ensuite, pour visualiser le résultat, il suffit de désactiver les couleurs du site dans l'option *Accessibilité*. Pour ne pas obliger les utilisateurs à changer les couleurs afin d'obtenir plus de contraste, le site doit afficher des couleurs bien distinctes pour le fond et le contenu, comme le précise le point à contrôler 2.2 des recommandations WCAG 1.0. Cependant le contraste entre deux objets peut être un élément difficile à appréhender. C'est pourquoi des logiciels tels que AccessColor [159], Accessibility Color Wheel [219] ou encore Color Constrast Analyser [210] permettent de calculer si le contraste est suffisant. L'outil Color Blindness Simulator [189], est quant à lui spécialisé dans la simulation de l'affichage de la page de la manière dont les personnes handicapées visuelles la perçoivent.

4. la désactivation des images et des scripts
Les scripts et les images peuvent également être désactivés. Les directives de la WAI préconisent d'utiliser des textes alternatifs pour décrire les images. Le navigateur Opera possède une option *Images* dans le menu *Affichage* permettant de désactiver les images. Le développeur peut utiliser cette fonction pour vérifier que toutes les images possèdent un texte alternatif et que l'information transmise par ce texte est identique à l'information donnée par l'image. Lynx étant un navigateur en mode texte, il n'affiche pas les images, ce qui permet de vérifier que les textes alternatifs restituent bien toute l'information. Par ailleurs Internet Explorer propose dans ses options de

désactiver les scripts. Cela permet de vérifier que l'information est toujours présente, même sans les scripts, c'est-à-dire qu'une alternative a été prévue pour remplacer le script ou que ce dernier ne véhiculait pas d'information importante.

Pour tester un site web, des barres d'outils peuvent être ajoutées aux navigateurs. Le logiciel Web Accessibility Toolbar [158] de l'équipe AIS (Accessible Information Solution) permet d'ajouter une barre d'outils à Internet Explorer. Le navigateur Mozilla Firefox propose lui aussi une barre d'outils, nommée Accessibar [224]. Celles-ci permettent de tester l'accessibilité des pages web et proposent notamment de désactiver les feuilles de style, les images, de modifier les couleurs et de passer en niveaux de gris. Web Accessibility Toolbar donne également un lien qui permet de visualiser la page en cours avec Lynx Viewer. Il est possible aussi d'observer le site de la même manière que peuvent le voir certains handicapés visuels (par exemple atteints de glaucome, de daltonisme, de cataracte…). Le tableau 1.6 permet de comparer les différentes fonctionnalités offertes par ces différentes barres d'outils.

De plus, d'autres outils permettent de tester les recommandations officielles propres à un pays. En France, les recommandations Accessiweb peuvent être testées avec la barre d'outil Accessiweb [170] à intégrer à Internet Explorer. Elle a été créée par l'association BrailleNet, en collaboration avec l'équipe de AIS.

Fonctionalités	AIS pour IE	Accessibar pour Firefox
Liens vers les recommandations W3C		
HTML 4.01		*
XHTML 1.0 et 1.1		*
CSS 1 et 2		*
WCAG 1.0	*	*
Techniques WCAG 1.0	*	
Liens vers des guides sur l'accessibilité		
Livre Blanc Braillenet	*	
Guide Braillenet	*	
Plongez dans l'accessibilité	*	
Liens vers des outils de validation en ligne		
HTML	*	*
CSS	*	*
The Wave	*	
Bobby	*	*
Les images et scripts		
Désactiver les images	*	*
Désactiver le Java	*	*
Désactiver le javascript	*	*
Afficher la dimension des images	*	*
Afficher le poids des images		*
Entourer les images sans attribut alt	*	*
Entourer les images avec attribut alt vide	*	*
Entourer les images sans dimension		*
Les feuilles de styles et les couleurs		
Désactiver	*	*
Ajouter une feuille de style personnalisée	*	*
Désactiver les couleurs	*	*
Outils d'analyse des contrastes	Juicy Studio	
Simulations		
Handicaps visuels	*	
Visualiser avec Lynx	*	
Structure		
Accesskey	*	*
Index de tabulation	*	*
Afficher les blocs	*	*
Afficher la taille des blocs		*

Tab. 1.6 – Comparaison de différentes barre d'outils s'ajoutant aux navigateurs.

1.5 Conclusion

L'utilisation des e-services administratifs est perçue comme une amélioration par 63% des personnes handicapées interrogées. Malgré tout encore 40% n'utilisent pas les e-services essentiellement à cause de difficultés d'utilisation [116][117]. Peu de sites web publics sont effectivement accessibles. Une étude réalisée en 2007 [155] précise que 5,3% des sites web publics en Europe possède le niveau A d'accessibilité (évaluation réalisée manuellement). Les résultats sont encore plus alarmants concernant les sites commerciaux puisqu'aucun n'est A-accessible[9]. De nombreuses lois imposent l'accessibilité aux sites web publics, mais si des améliorations sont constatées (3% en 2005 à 5,3% en 2007 de sites web publics européens A-accessible), les résultats significatifs se font toujours attendre. Actuellement face à ces nouveaux e-services qui leur sont inaccessibles, les personnes handicapées se trouvent encore en marge de la société. Les sites administratifs non accessibles ne font que renforcer la fracture sociale entre les personnes valides et les personnes handicapées.

Pour expliquer le nombre important de sites web non accessibles, les webmestres évoquent les raisons suivantes : l'absence de formation en accessibilité, des coûts de développement *a priori* trop importants, le manque d'intérêt pour ces profils d'usagers, la crainte d'appauvrir le site [78]. Or, comme nous l'avons présenté, ces idées sont fausses mais subsistent. Pour que les webmestres révisent leur jugement, il faut mettre les moyens nécessaires à la diffusion d'informations réelles sur l'accessibilité.

Au terme de ce chapitre, nous conclurons en rappelant les avantages à créer un site web accessible :

- Répondre aux obligations légales.
  Dans tous les pays, les exigences d'accessibilité sont de plus en plus fortes et les gouvernements commencent à légiférer dans ce sens : les lois en Italie, en France, etc., et la création du consortium EuroAccessibility en Europe pour la mise en place d'une autorité européenne de certification de l'accessibilité des sites web... Ainsi, rendre un site accessible permet de répondre aux obligations législatives qui deviendront incontournables dans quelques années en France mais aussi dans tous les pays européens.
- Réduire les coûts associés à la maintenance.
  Le respect des standards de l'accessibilité rend la maintenance des sites web plus facile et donc moins coûteuse. Par exemple, l'utilisation des feuilles de style pour séparer le fond et la forme permet de changer le design du site rapidement sans avoir à modifier toutes les pages.
- Augmenter la clientèle.
  En France, 3,1 millions de personnes déclarent une atteinte visuelle. Selon l'O.M.S, en 2002 ce chiffre s'élevait à 15,5 millions de personnes en Europe, directement concernées par le besoin de sites web accessibles. Avec l'ensemble des seniors (ayant des difficultés de lecture, des tremblements du bras...), c'est plus de 100 millions de personnes en Europe qui sont directement concernées. De plus, toutes les personnes n'utilisant pas de navigateurs standards (par exemple les personnes voyageant beaucoup et utilisant

[9]Les résultats relevés avec un validateur automatique sont différents et moins stricts : 12,5% de sites publics et 3,9% de sites commerciaux européens A-accessibles en 2007.

des technologies mobiles d'accès à Internet) pourront naviguer sur un site accessible. L'accessibilité d'Internet ne concerne donc pas seulement les personnes handicapées, mais toute personne ayant des besoins ou des modes d'utilisation d'Internet moins conventionnels. Dès lors, rendre un site accessible permet d'augmenter le nombre d'internautes naviguant sur le site et, par conséquent, le nombre de clients potentiels s'il s'agit d'un site commercial.

- Améliorer l'intégration des personnes handicapées.
  L'accessibilité constitue effectivement un facteur d'intégration sociale, professionnelle et culturelle important pour les personnes handicapées.

Chapitre 2

Se repérer/se diriger dans un site et s'exprimer à l'écrit

Nous nous intéressons dans ce chapitre aux solutions pour parer aux difficultés que peuvent rencontrer les personnes handicapées sur Internet. Dans un premier temps, ce chapitre traitera du problème de la navigation au sein d'un site web pour atteindre l'information désirée le plus rapidement possible. Puis, dans un deuxième temps, nous considérerons les difficultés que peuvent rencontrer les personnes à mobilité réduite pour saisir un texte, par exemple dans un formulaire.

2.1 Introduction

Comme nous l'avons vu dans la section 1.4.1 (page 38), un site web utilisable doit être efficace et efficient. Mesurer l'efficience correspond à vérifier que le but de l'internaute a été atteint avec un minimum d'efforts et en un minimum de temps. Un site web ergonomique diminue donc le temps de recherche d'un internaute par rapport au même site dans une version non ergonomique [26]. On parle notamment de la « règle des 3 clics » selon laquelle toute information d'un site doit être disponible en maximum 3 clics de souris. Cette règle est évidemment difficilement applicable pour les sites web volumineux. Néanmoins, d'après une étude de J. Porter [233], cette règle n'est pas fondée et rien ne justifie son existence puisque l'important n'est pas le nombre de clics pour trouver l'information désirée mais de trouver l'information. Retenons finalement qu'elle permet de rappeler plus généralement l'importance de minimiser les difficultés que les utilisateurs vont rencontrer pour trouver leur

information. Un critère primordial en termes d'ergonomie des sites web concerne le guidage. Il s'agit de l'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur vers son but. Nous nous attarderons sur l'un des sous-critères du guidage qu'est « l'incitation ». Cela concerne les liens, les mécanismes de recherche d'information et la navigation. L'utilisation d'index et de tables des matières est une façon de guider l'utilisateur sur un site web. Pour résoudre ce problème général de navigation, des réponses ont été apportées : elles visent à réduire le nombre d'internautes qui se perdent au sein d'un site web et qui préfèrent de ce fait abandonner leurs recherches (ce qui est souvent le cas chez les personnes qui utilisent un lecteur d'écran [131]). La solution, qui consiste principalement en l'utilisation de techniques d'aide à la navigation par visualisation d'un plan du site [15], fera l'objet de la section 2.2 de ce chapitre.

Nous nous sommes également intéressés au cas des personnes ayant des facultés motrices très réduites qui se servent d'aides techniques pour utiliser leur ordinateur et de ce fait accéder à Internet. D'une part, certaines personnes handicapées motrices sont aptes à utiliser une souris d'ordinateur classique tandis que d'autres seront plus à l'aise avec un système de pointage tel qu'un track-ball ou une plaque sensible. Reste que ces dispositifs ne sont pas adaptés aux personnes ayant des problèmes de coordination. Dans ce cas, il est possible d'utiliser un joystick ou une manette à la place d'une souris et de les configurer à l'aide d'un logiciel tel que JoyMouse [249]. D'autre part, pour certaines personnes l'utilisation du clavier peut également s'avérer difficile. Afin de pallier aux différents handicaps existants, de nombreux systèmes existent. Ce sont ces systèmes d'aide à la saisie qui seront présentés dans la section 2.3.

2.2 Faciliter l'orientation dans un site grâce au plan du site

Le grand nombre de pages sur les sites web a contribué à augmenter les problèmes de navigation. De plus, et ce malgré la présence d'un sommaire présentant la structure du site, de nombreux liens sont souvent insérés dans le contenu des pages créant ainsi des raccourcis vers d'autres pages du site. Face à cette multitude de liens et de pages web accessibles par la page d'accueil d'un site web, les internautes se trouvent souvent désorientés [91] dès lors qu'il s'agit de trouver l'information désirée dans un site web. Dans la majorité des cas, cela les conduit à ne pas trouver ce qu'ils veulent [227]. Ce manque de repères est d'autant plus problématique pour les personnes handicapées visuelles qui doivent se faire une représentation mentale du site. Avant de choisir une destination il faut en effet avoir en tête les autres possibilités. Une quantité trop importante de liens, et des pages mal organisées entre elles peuvent s'avérer difficilement mémorisables pour un internaute non-voyant. Face à cette complexité, les webmestres eux-mêmes peuvent rencontrer des difficultés à se représenter la structure globale de leurs sites. Lors de l'ajout d'une nouvelle page dans un site web, il est primordial de bien connaître la structure de ce dernier afin de placer la nouvelle page à bon escient. Dans le cas contraire, un ajout mal positionné d'une nouvelle page contribuera à rendre la navigation au sein du site encore plus difficile.

Pour naviguer facilement, ou pour insérer au bon endroit une page, il est nécessaire de bien connaître la structure du site web. Certains sites web proposent effectivement une page réservée à l'affichage de leurs plans. C'est ce point qui sera étudié dans la section 2.2.1. Toutefois, encore beaucoup de sites n'en disposent pas. On trouve également des outils de génération de plan de site, qui seront présentés dans la section 2.2.2. Il convient de préciser que les représentations sont variées, qu'il s'agisse d'un plan de site intégré au site web ou d'un outil de génération de plan.

2.2.1 Les pages web dédiées à l'affichage d'un plan du site

Face aux nombreuses pages proposées depuis la page d'accueil des sites web, un internaute peut rencontrer des difficultés à repérer directement la page ou la direction qui l'intéresse. Certains sites mettent alors à disposition une page spéciale contenant uniquement le plan du site. Pour faciliter la navigation au sein d'un site, il ne tient qu'au webmestre de mettre en ligne ce sitemap et de le maintenir à jour.

Ces plans de sites représentent une vue statique du site web. En ce sens la représentation du site est identique pour tous les internautes et il n'est pas possible de la personnaliser. De plus, cette représentation est générée par le webmestre qui décide lui-même de la précision du plan — il peut par exemple décider de ne présenter que les deux premiers niveaux de profondeur des pages de son site, et occulter ainsi certaines pages qui sont plus profondes dans la structure. Le niveau de détail retenu dépend du site web et a fortiori de son nombre de pages. Si le site est volumineux en terme de pages, le plan devra être établi sur le principe d'une table d'orientation, et ne devra pas être exhaustif (car cela pourrait entraîner le

contraire de l'effet désiré). Si le contenu est abondant et que le webmestre souhaite donner un niveau de détail important, l'affichage du plan peut se faire de manière échelonnée (action plier/déplier). Ainsi l'internaute ne sera pas désorienté face à l'ampleur du plan et pourra progressivement entrer dans le détail.

Pour générer ce plan de site, le webmestre peut avoir recours à différentes méthodes, qui dépendent entre autres de la manière dont le site web est administré. Dans un premier temps, on peut observer des sites web « statiques » où le webmestre a pour rôle de coder les pages HTML. Dans ce genre de site, seules les personnes qui détiennent des connaissances en informatique et en programmation sont à même d'effectuer le maintien et les mises à jour du site. Dans ces circonstances, le webmestre doit rédiger une à une les différentes pages. Cela s'applique également à la page contenant le plan du site. Pour des sites possédant de nombreuses pages et s'enrichissant souvent de nouvelles pages, le webmestre doit vérifier que le plan correspond bien à la réalité actuelle du site, et si besoin le compléter ou le modifier. Le webmestre est alors responsable du site web, de la création à la maintenance, en passant par la mise à jour de contenu (pages et plan du site). La gestion de ce genre de site s'effectue de manière centralisée, ce qui apporte une charge de travail considérable pour le webmestre. Cela conduit à de nombreux problèmes, comme les goulots d'étranglement de la production web — c'est-à-dire une mise à jour fréquente des pages et du plan de site —, le manque d'implication des personnes qui pourraient participer à la vie du site web, et les risques d'erreurs liées à la rédaction de contenu dont le webmestre ne connaît pas tous les aspects. Dans un second temps, les systèmes de gestion de contenu (*Content Management System*, CMS) permettent de gérer les sites de manière décentralisée. La gestion de contenu permet à n'importe quel membre d'une organisation ou d'une communauté de mettre en ligne de l'information sans difficulté technique. A cette fin, la personne se contente de saisir des contenus, généralement *via* un formulaire, par injection directe de documents produits avec les outils bureautiques quotidiens, quitte à automatiser les conversions de formats nécessaires. Le contenu produit par un utilisateur est stocké dans une base de données. Il est ainsi accessible et modifiable par les utilisateurs autorisés. Certains CMS proposent également de créer automatiquement le plan du site (après avoir indiqué la présentation désirée). On peut notamment penser aux CMS SPIP [240] (intégrant une application de plan de site) et Typo3 [244] (proposant une application de plan de site sous forme de plugins). L'avantage des CMS en ce qui concerne le plan du site est que ce dernier change automatiquement en fonction des modifications apportées à la structure du site. Aucune correction manuelle du plan du site n'est nécessaire lors de la réorganisation du site web ou lors de l'ajout de nouvelles pages.

Que le plan du site soit généré manuellement par le webmestre ou automatiquement par le CMS, il peut prendre différentes formes. Il y a quelques années, avec la naissance de nouvelles technologies comme le flash, un site web agréable se devait d'être très coloré, animé, et imagé. Cette tendance s'est également retrouvée dans l'apparence des plans de sites. Il était en effet courant de se trouver face à des plans de sites avec des formes diverses et variées (allant de l'arbre classique à des formes plus arrondies), et comportant de nombreuses animations apparaissant par exemple au survol de la souris. La frénésie du départ étant passée, et se souciant de plus en plus de la clarté des informations et de leur accessibilité, les

webmestres ont désormais redonné aux sites web un aspect beaucoup plus épuré. Les plans de sites web actuels se sont eux aussi simplifiés, en se présentant désormais principalement sous forme de liste de liens.

La figure 2.1 représente le plan du site du Sénat[1]. Il a l'apparence d'une table des matières contenant des liens HTML. C'est une structure classique que l'on retrouve actuellement dans la plupart des sites web.



FIG. 2.1 – Plan du site du Sénat.

Dans beaucoup de sites web on observe des plans présentés à l'aide de tableaux HTML (ou utilisant une disposition sous forme de tableau, avec des cases les unes à côté des autres). Les figures 2.2 et 2.3 montrent des exemples de ce type de présentation en tableau, visibles respectivement sur le site « Service Public »[2] et sur le site « Apple »[3]. Remarquons que le plan du site de la figure 2.2 ne présente qu'un ensemble restreint de pages. Le trop grand nombre de pages empêchant de réaliser un plan de site exhaustif et restant facilement compréhensible par les internautes, seuls les deux premiers niveaux de pages sont représentés dans ce plan.

D'autres sites web comme celui de « France 5 »[4], possèdent un plan de site hybride (représenté par la figure 2.4 (a)). Il mélange une partie sous forme de liste et une partie sous forme de tableau. Sur cet exemple, on peut voir dans la première partie sous forme de liste, les liens en noir suivi d'un résumé en gris. Dans la seconde partie sous forme de tableau, les liens sont en gris, et il est difficile de voir rapidement où commence un nouveau lien. La figure 2.4 (b) présente la même page où nous avons entouré les liens. La présentation des liens de ce plan n'est pas homogène, ce qui rend la lecture difficile et réduit l'utilité du plan.

[1] `http://www.senat.fr/plan.html` - dernier accès le 09/05/2008
[2] `http://www.service-public.fr/aide/plan.html` - dernier accès le 09/05/2008
[3] `http://www.apple.com/fr/sitemap/` - dernier accès le 09/05/2008
[4] `http://www.france5.fr/plan/` - dernier accès le 09/05/2008

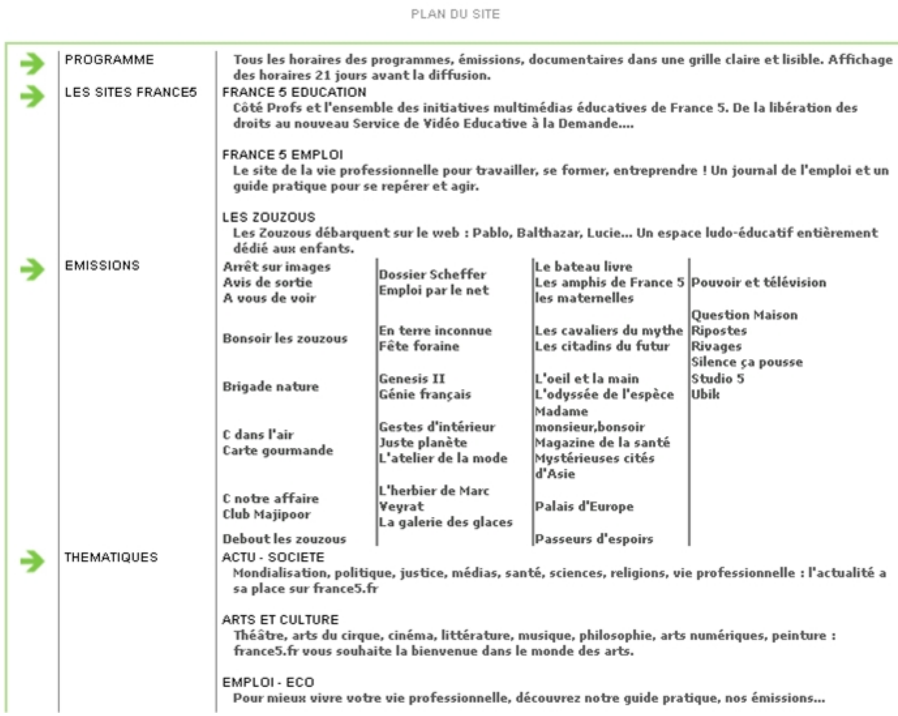


Fig. 2.2 – Plan du site du portail de l'administration française : Service-Public.fr.

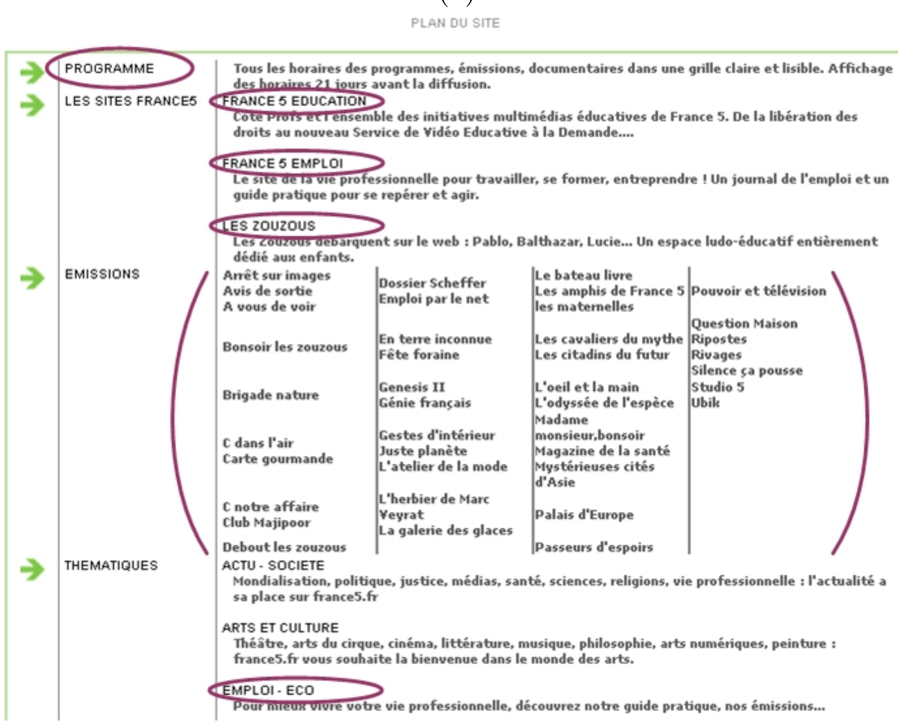


Fig. 2.3 – Plan du site Apple.

Nous constatons que certains sites ne requièrent pas forcément de plan. Si toutes les pages sont en effet accessibles en un clic depuis la page d'accueil de façon évidente (depuis le menu par exemple et non par des liens disséminés dans le texte), le plan du site n'est pas obligatoire. La page d'accueil joue, dans ce cas, le rôle du plan du site. Cela concerne uniquement les « petits » sites web qui ne possèdent qu'un seul niveau de profondeur. Pour les « gros » sites, comme les sites web commerciaux ou les sites web publics, les plans sont très importants. Ils permettent de faciliter la navigation au sein d'un site web en proposant à l'internaute une alternative au sommaire. Cependant, une étude menée en 2006 sur 73 sites publics français [55] a montré que 1/3 d'entre eux ne proposent pas de plan de site. Remarquons que même lorsqu'ils existent, ces derniers ne sont pas personnalisables : la présentation, l'exhaustivité, et la mise à jour du plan restent uniquement du ressort du webmestre.



(a)



(b)

FIG. 2.4 – Site de France 5 : plan original (a) et repérage des liens sur ce même plan (b).

2.2.2 Les générateurs de plans de sites

Beaucoup de sites ne se soucient guère des problèmes d'accessibilité et ne proposent donc pas nativement de plan de site. Pour se repérer plus facilement dans l'arborescence d'un site qui ne contient pas de plan, il peut être pratique pour les malvoyants, les non-voyants et plus généralement pour tout internaute, de pouvoir générer rapidement le plan du site et de l'utiliser pour accéder aux différentes pages. Mais ces outils de génération de plans peuvent aussi servir aux webmestres qui désirent avoir un aperçu de la structure de leur site. Il existe différents types de structures de site web : séquentielle, hiérarchisée, en réseau... Cependant, rares sont les sites pouvant se baser sur une seule de ces structures. Face à des structures complexes, les générateurs de plan peuvent grandement faciliter le travail des webmestres.

WebTracer [228] est un outil développé par la société Nullpointer. En donnant l'adresse d'un site, il permet de représenter tous les liens de ce site ainsi que toutes les pages, documents pdf, feuilles de styles et documents word... De plus, pour chaque nœud du graphe, il est possible d'obtenir des détails tels que son URL, son titre, sa taille en octets et le nombre de liens qu'il contient, simplement en survolant le nœud avec la souris. Cette représentation autorise l'agrandissement, la réduction, la rotation et indique le nom de la page sur laquelle est placé le pointeur. WebTracer est un logiciel s'exécutant en deux étapes : l'étude du site et la visualisation de la carte. La figure 2.5 représente le plan du site de Polytech'Tours généré par WebTracer. Malgré ce rendu graphique agréable, il est possible de faire quelques reproches à ce logiciel. A savoir la représentation de tous les liens et de toutes les pages et documents garantit un plan complet mais implique un grand nombre de nœuds ajoutés qui surchargent le graphe mais qui n'apportent pas réellement d'information utile.

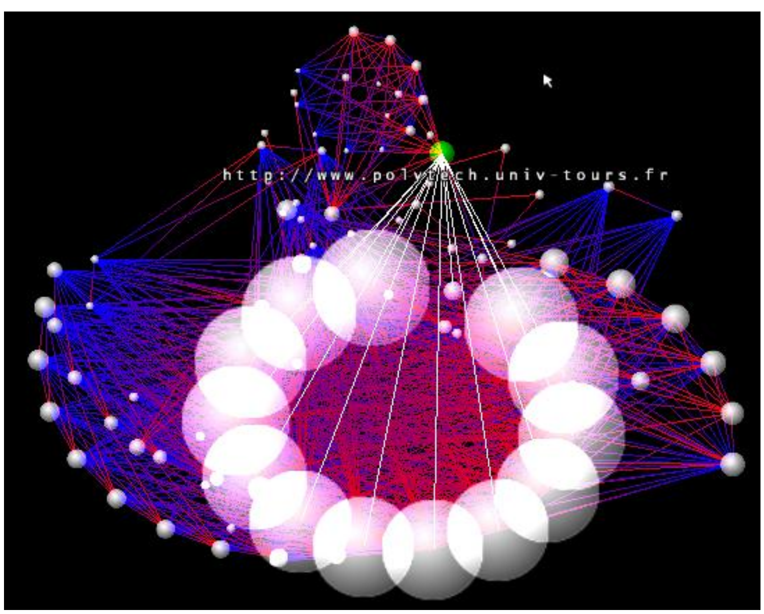


FIG. 2.5 – Site de Polytech'Tours représenté avec WebTracer.

VizServer [207] de la société Inxight permet de visualiser des collections sous trois formes différentes : Star Tree, Time Wall, et TableLens. Star Tree propose une représentation très claire d'un site sous la forme d'un œil et offre la possibilité d'afficher ou non les sous-pages d'une page. Des démonstrations d'exécution de Star Tree et de TableLens sont disponibles en ligne.

L'outil GEOmi [226] (Geometry for Maximum Insight, c'est-à-dire « géométrie pour maximiser la perspicacité ») est développé par l'institut de recherche australien NICTA. Bien que la génération d'arborescence de site web ne soit pas sa première utilité, le plugin « HTML Graph Generator » fournit cette possibilité. La figure 2.6 présente le résultat obtenu sur le site du Conseil Général d'Indre-et-Loire[5] (CG37). On peut observer que très peu de nœuds sont représentés. Ici, seules les pages HTML sont affichées. Les autres fichiers (pdf, word, css, vidéo. . .) n'apparaissent pas. Cependant beaucoup de pages web ne sont pas représentées non plus. De plus, tous les noms des nœuds sont indiqués à côté des nœuds, ce qui surcharge très vite l'affichage, surtout lorsque les titres des pages sont longs. Ajoutons qu'une rotation du graphique est possible. La figure 2.7 présente le même plan que sur la figure 2.6, après rotation et on y observe un problème d'affichage : les nœuds et leurs noms ne sont pas liés. Ce sont en fait deux plans différents qui ne pivotent pas en même temps. La rotation est donc inutilisable.

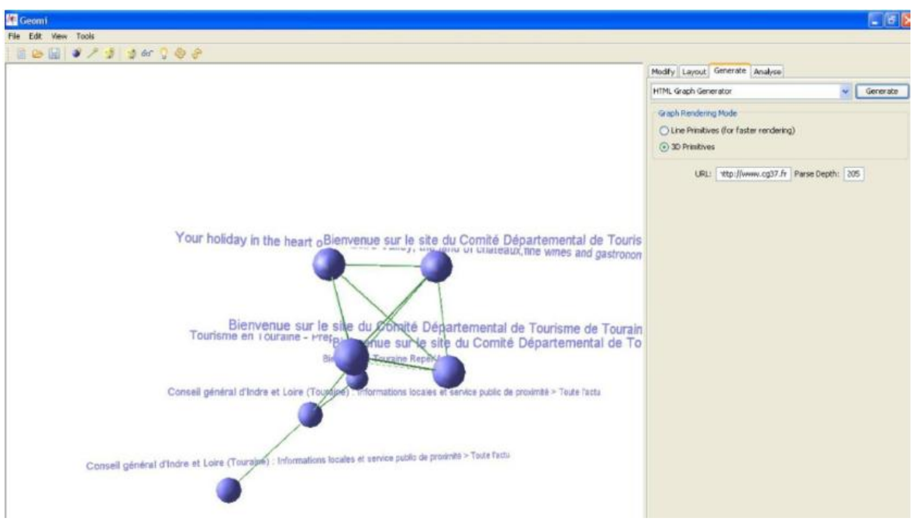


FIG. 2.6 – Site du CG37 représenté avec GEOmi.

L'outil WebTOC [98] est un programme Java permettant de visualiser la structure d'un site web sous la forme d'une table des matières. Ce mode d'affichage a été choisi afin de proposer une interface intuitive (identique à celle de l'explorateur de Windows) et facile à prendre en main. Dans cet outil, tous les documents (texte, image, audio. . .) sont représentés avec des couleurs différentes afin de les distinguer rapidement. Précisons que WebTOC est composé de deux parties. Premièrement, un analyseur syntaxique (parser) génère la table

[5] `http://www.cg37.fr` - dernier accès en juin 2006

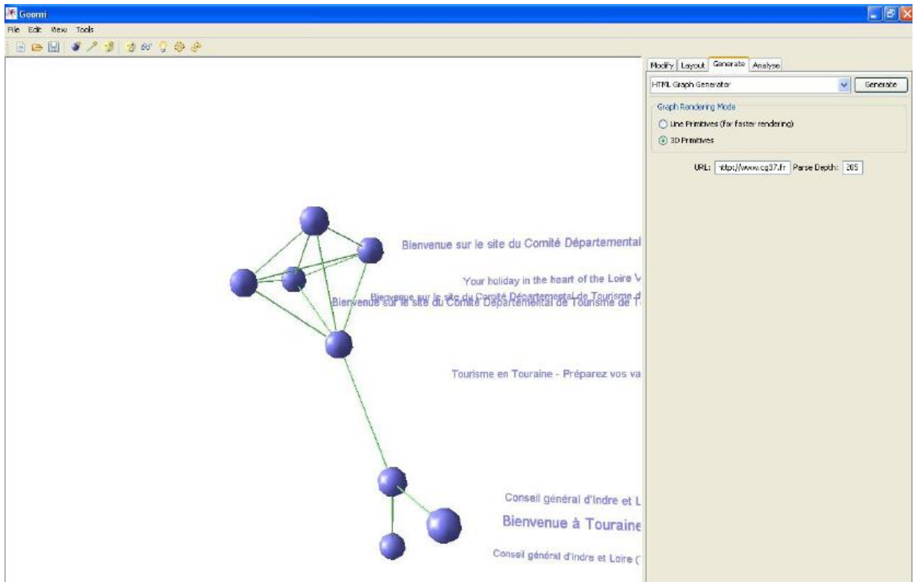


Fig. 2.7 – Site du CG37 représenté avec GEOmi après rotation.

des matières. Seuls les liens correspondant au plus court chemin pour joindre chaque page à la page d'accueil sont conservés. Deuxièmement, une interface utilisateur (applet java) permet à l'utilisateur d'interagir avec son navigateur web en y affichant la page souhaitée d'un simple clic sur le plan généré.

L'outil CCT [47] (Collapsible Cylindrical Trees) propose quant à lui une visualisation en forme de cylindre 3D pour les deux premiers niveaux et sous forme de listes pour les niveaux suivants. Cet outil propose lui-aussi une réelle interaction avec l'utilisateur comme par exemple l'accès aux pages du site en cliquant sur le plan. Au départ, uniquement les deux premiers niveaux sous forme de cylindres sont affichés. C'est ensuite à l'utilisateur de demander le dévoilement des niveaux inférieurs. Cet affichage en deux temps permet d'augmenter l'utilisabilité et les performances du plan du site [67], notamment pour les sites web volumineux.

Navibar [237] est une extension pour Firefox permettant de représenter le plan d'un site web sous forme d'une table des matières dynamique dans un panneau latéral au navigateur (figure 2.8). Cette représentation est navigable, c'est-à-dire que chaque nœud de l'arbre peut conduire à l'affichage de la page dans le navigateur. Pour l'internaute, la navigation au sein d'un site se trouve alors centralisée en un seul endroit indépendant du site web. Cette extension a besoin pour fonctionner d'un plan de site source sur le serveur mis en place par le webmestre du site. Ce plan de site source se présente sous la forme d'un fichier `rdf` contenant la structure du site, le titre et l'URL de chaque page. Le webmestre doit créer ce plan de site manuellement [229] et le maintenir à jour.

Sur le même principe que Navibar (à savoir l'utilisation d'un plan source) mais dans un tout autre but, United Sitemaps (appelé autrefois Google Sitemaps) recommande aux webmestres de créer une sitemap, aussi appelée googlemap. Le but de ce plan source n'est pas de

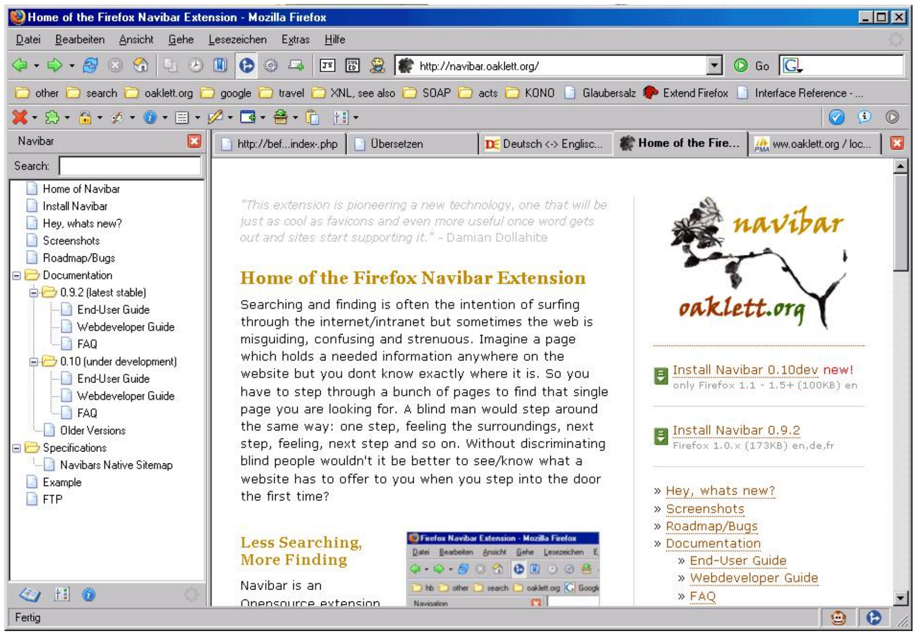


Fig. 2.8 – Barre de navigation Navibar.

générer un plan de navigation comme pour Navibar, mais d'indiquer facilement aux moteurs de recherche les pages de leurs sites à explorer. Habituellement, pour indexer un site web, les robots d'exploration parcourent les différentes pages du site qu'ils trouvent en utilisant les liens internes au site. Chaque page du site web est analysée et, selon les liens qu'elle contient, conduit à l'analyse d'autres pages. Or ce processus s'avère extrêmement long pour les sites web volumineux. Dès lors, pour améliorer la vitesse de soumission d'indexation d'un site web et également pour rendre les moteurs de recherche plus réactifs aux actualisations, Google propose aux webmestres de réaliser une sitemap [238]. C'est le webmestre qui indique aux robots où aller, c'est-à-dire quelles sont les pages à indexer. Pour cela, une Google sitemap recense dans un fichier `xml` l'ensemble des URL du site web. D'autres informations facultatives peuvent y être ajoutées comme la date de dernière modification, la fréquence probable des modifications, et la priorité d'une URL par rapport aux autres pages du site. La figure 2.9 représente un exemple de sitemap Google contenant deux pages web. Précisons que chaque Google sitemap peut lister au maximum 50 000 URL. Si le site web contient plus de 50 000 URL, il est possible de générer plusieurs sitemap et de les répertorier dans un fichier d'index sitemap.

Des générateurs automatiques sont capables de créer cette googlemap. Le générateur SitemapDoc [151] peut générer la googlemap de site web contenant au maximum 500 pages. Il est utilisable en ligne. Il propose, en plus du format xml, un format HTML, ne contenant que les URL et éventuellement une description de chaque page. Cette googlemap convertie en HTML peut servir de base à la création d'un plan de site web. Remarquons que la structure réelle du site sera à faire manuellement puisque la googlemap ne correspond qu'à une liste de liens non hiérarchisés. On trouve également des générateurs sous forme de script,

```
<?xml version= "1.0" encoding= "UTF-8"?>
<urlset xmlns= "http ://www.google.com/schemas/sitemap/0.9">
<url>
<loc>http ://www.monsite.fr/</loc>
<lastmod>2007-12-12</lastmod>
<changefreq>monthly</ changefreq >
<priority>0.8</ priority >
</url>
<url>
<loc>http ://www.monsite.fr/page1.html</loc>
<lastmod>2007-12-20</lastmod>
< changefreq > weekly</ changefreq >
<priority>0.8</ priority >
</url>
</urlset>
```

Fig. 2.9 – Exemple de Google Sitemap au format xml.

comme Google Sitemap Gen [199], écrit en python et s'exécutant sur le serveur web du site dont on veut réaliser la googlemap, et des générateurs sous forme de logiciel à installer en local comme YoodaMap [161]. Une fois le fichier sitemap créé et transféré à la racine du site (ou à la racine des pages à indexer contenues dans le sitemap), il faut soumettre ce fichier au moteur de recherche (par le biais de l'interface de transmission du moteur de recherche ou en indiquant l'emplacement du sitemap dans un fichier robots.txt). Ainsi, en créant une googlemap avec le logiciel SitemapDoc, le webmestre favorise l'indexation de son site web et peut générer une trame HTML pour aider à la navigation dans son site web. Ce logiciel s'oriente bien évidemment côté webmestre pour la partie indexation du site, mais également pour la partie génération du plan HTML. Ce plan n'est effectivement pas finalisé (seule la liste non organisée des pages est présente au départ), et il a pour vocation d'être mis en ligne par la suite, comme tout autre page du site.

SortSite [188] est un outil de la société Electrum permettant d'analyser un site web (liste des liens cassés, erreurs dans le code...). Il coûte de 129 à 500 dollars selon les versions, mais est disponible en version d'essai de 30 jours (100 pages maximum testées). Il permet de générer un plan de site pour repérer où se situent les erreurs. La figure 2.10 illustre le plan généré par SortSite (en version d'évaluation) sur le site de Polytech'Tours en janvier 2008. Chaque page est représentée par une tour plus ou moins large selon sa distance par rapport à la page d'accueil. Les tours les plus hautes représentent les pages qui comportent le plus d'erreurs (avec en rouge les erreurs critiques et en orange les warnings).

PowerMapper [187] est un générateur de plan de site web. Cet outil coûte de 150 à 500 dollars selon les versions, mais est disponible en version d'essai de 30 jours. PowerMapper propose plusieurs styles d'affichages différents. La figure 2.11 illustre plusieurs types d'affichages du plan du site de Polytech'Tours en janvier 2008 : chaque page est représentée par une tour de plus en plus petite lorsque l'on s'éloigne de la page d'accueil (a), par regrou-

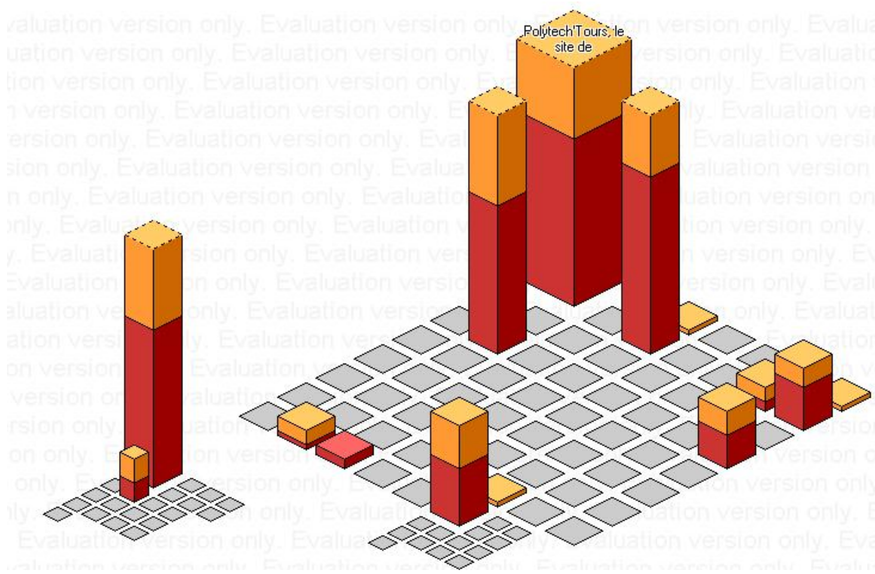


FIG. 2.10 – Site de Polytech'Tours analysé par SortSite.

pement de pages en paquet (b), grille en deux dimensions (c) ou en trois dimensions (d) (avec 3 niveaux de profondeur), en arbre (e) ou encore sous forme de table des matières (f). Il est également possible d'exporter le plan au format xml de google sitemap. Les plans sont interactifs : par exemple dans la visualisation par regroupement des pages (figure 2.11 (b)), il est possible de cliquer sur l'icône « + » se trouvant à côté de l'image représentant une page pour que celle-ci se trouve centrée sur le plan à la place de la page d'accueil. De plus, les pages du plan généré sont cliquables, mais la visualisation se fait dans le logiciel PowerMapper et non dans un navigateur web standard. Ce logiciel est donc plutôt orienté webmestre, d'autant que la génération du plan est très longue : plus de 4h pour le site de Polytech'Tours où 632 fichiers (pages, images ou pdf) ont été détectés.

Ixsite Web Analyzer [208] de la société Ixacta est un outil de management de sites web qui permet de détecter les liens cassés, les images manquantes... Il génère des rapports et des statistiques ainsi que des diagrammes pour représenter les relations entre les fichiers du site web (avec trois niveaux de profondeur), les liens sortants et entrants pour une page, la distance entre les pages (sous forme de cercles centrés sur la page d'accueil). Sur la figure 2.12 on peut voir des illustrations représentant ces trois types de plans.

Hormis les représentations :
- simple, sous forme d'une table des matières (comme WebTOC et Navibar),
- sous forme d'un arbre 3D (comme GEOmi et WebTracer),
- sous forme de tours (comme SortSite),
- sous forme de cylindres couplés à un affichage en liste (comme CCT),

on peut trouver beaucoup d'autres possibilités : des représentations hyperboliques [74], sous forme d'arbres coniques [114], de murs en perspective [84], en ballons hyperboliques, avec des rectangles de couleurs différentes pour chaque niveau, et bien d'autres encore [67].

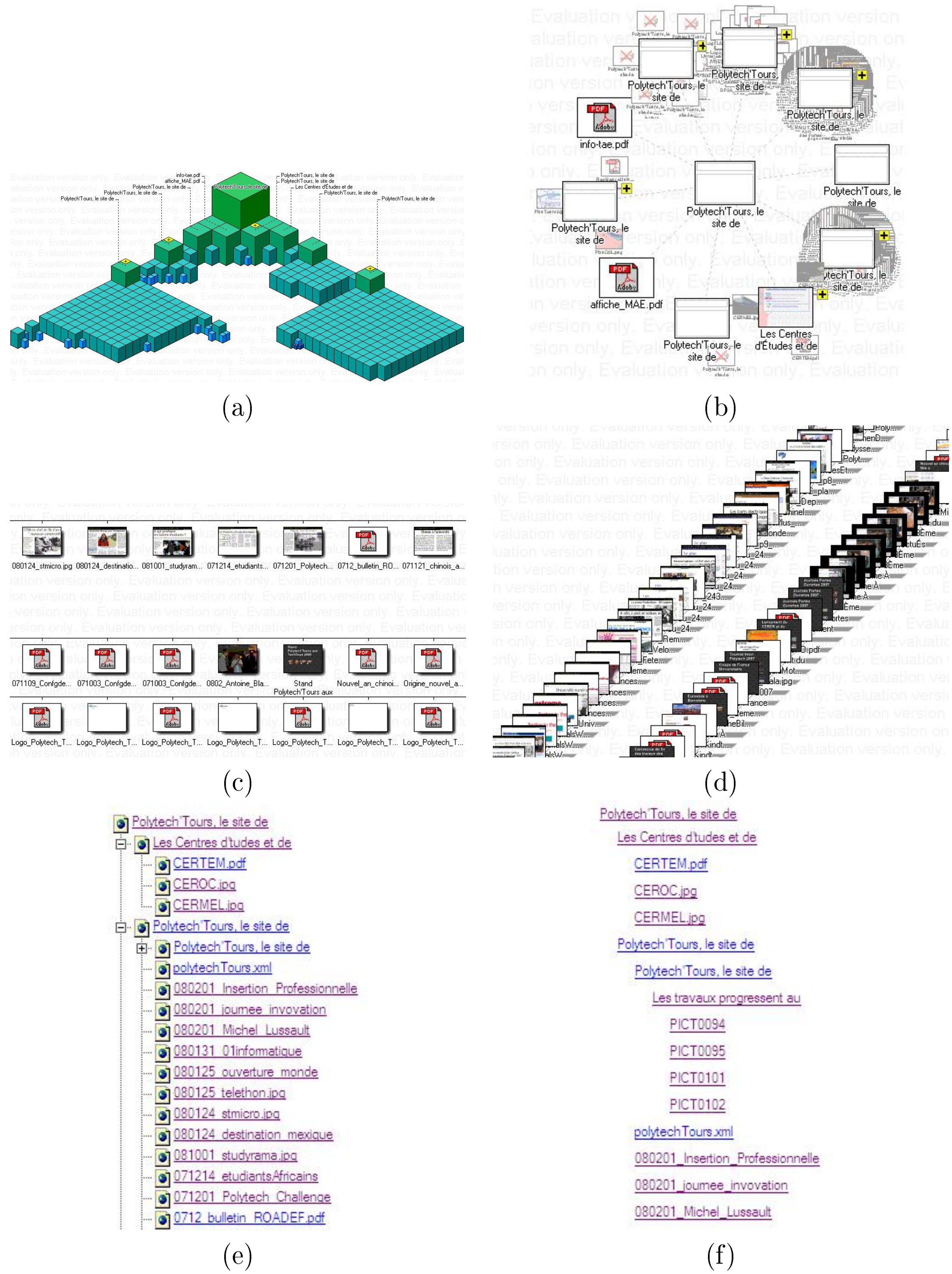
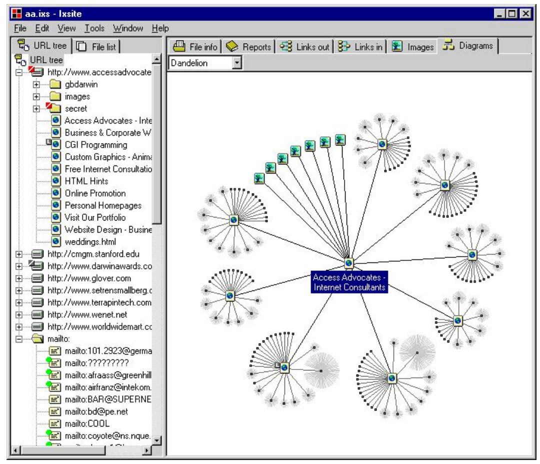
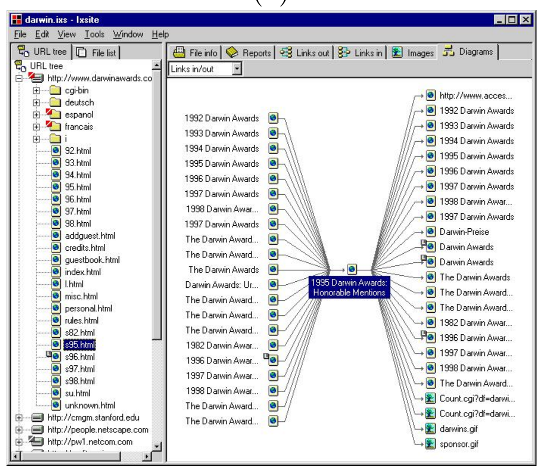


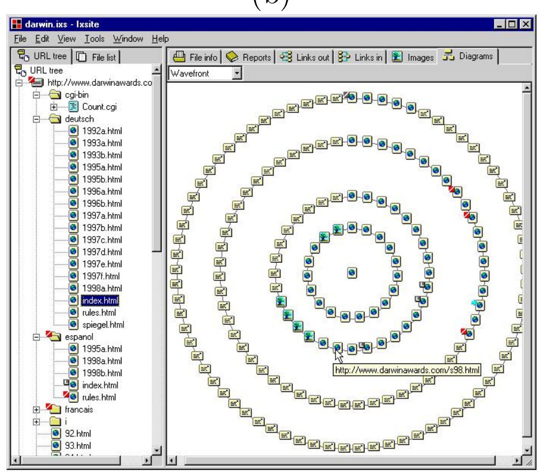
FIG. 2.11 – Affichages de plans du site de Polytech'Tours générés par PowerMapper : sous forme de tours (a), nuages (b), grille 2D (c) ou 3D (d), arbre (e), table des matières (f).



(a)



(b)



(c)

FIG. 2.12 – Illustrations des plans générés par Ixsite : relations entre les fichiers (a), liens sortants et entrants (b) et distance depuis la page d'accueil (c).

Le tableau 2.1 dresse un bilan des styles d'affichage des plans, et résume à qui est destiné le générateur et le plan.

	utilisation du générateur par	utilisation du plan par	style de plan
WebTracer	w	w	arbre 3D
GEOmi	w	w	arbre 3D
VizServer	w	w	oeil
SortSite	w	w	tour
Ixsite Web Analyzer	w	w	arbre 3D, cercles concentriques
PowerMapper	w	w	tour, grille, arbre, table des matières (statique ou dynamique)
Navibar	w	i	table des matières dynamique
SitemapDoc	w	i/robots	
Google Sitemap Gen	w	robots	
YoodaMap	w	robots	
CCT	i	i	cylindre
WebTOC	i	i	table des matières dynamique

TAB. 2.1 – Utilisateurs (**w**ebmestre ou **i**nternaute) des générateurs de plan et des plans eux-mêmes.

Les outils de génération de plans de sites tels WebTracer, VizServer, PowerMapper ou encore GEOmi, permettent d'obtenir une vue d'ensemble sur les éventuels agrégats de pages, sur l'équilibre global d'un site, et sont, de ce fait, plutôt réservé au webmestre soucieux de créer un site ergonomique. De plus, ce type d'outils générant des plans de sites sous forme d'images ou d'animations sont inexploitables par les personnes handicapées visuelles. A l'opposé, les outils WebTOC et Navibar, avec un affichage plus simple, s'orientent plutôt côté utilisateur. Ils proposent d'ailleurs une interaction entre le plan du site généré et le navigateur web de l'utilisateur. Notons que PowerMapper propose également un affichage sous forme de table des matières mais reste destiné au webmestre à cause de sa lenteur et de son affichage des pages web dans l'outil plutôt que dans le navigateur habituel de l'internaute. Navibar permet d'aider un internaute à naviguer mais ce plan ne peut être généré sans l'intervention du webmestre qui doit générer le plan source. Au lieu de demander aux webmestres de générer ce plan source (qui n'est d'ailleurs pas un standard web), il serait plutôt judicieux de les inciter à créer une page web dédiée à l'affichage du plan du site. Ainsi, cela profiterait à tous, et pas uniquement aux utilisateurs du navigateur Firefox. Concernant l'outil WebTOC, on peut d'ors-et-déjà remarquer son manque d'adaptation aux personnes handicapées. Force est de constater que :

- tous les documents sont représentés : cela peut encombrer le plan du site et l'affichage des images dans le plan du site ne présente aucun intérêt pour les personnes non-voyantes,
- les différents types de documents sont représentés par des couleurs différentes : la distinction ne peut pas se faire pour des personnes non-voyantes ou malvoyantes.

2.3 Faciliter l'écriture grâce aux aides techniques de substitution du clavier d'ordinateur standard

Les handicaps, ainsi que les niveaux d'atteintes, sont très variés et parfois évolutifs. Si certaines personnes handicapées n'éprouvent pas ou peu de difficultés à effectuer une certaine tâche, d'autres personnes plus atteintes ou ayant un autre type de handicap se trouvent désarmées. Les aides techniques permettent aux personnes handicapées de réduire les désavantages engendrés par le handicap.

Nous étudions dans cette section les aides techniques de substitution au clavier d'ordinateur standard. Plusieurs déficiences peuvent entraîner des impossibilités à utiliser un clavier traditionnel, comme par exemple des déficiences visuelles et des déficiences motrices. Une personne gravement malvoyante aura en effet des difficultés à voir les inscriptions sur les touches d'un clavier traditionnel, tandis qu'une personne handicapée des membres supérieurs aura des difficultés à appuyer sur les touches ou à se déplacer d'une touche à une autre. De ce fait, il n'existe pas une mais plusieurs aides techniques de substitution du clavier selon le handicap et le niveau d'atteinte. Celles-ci se répartissent en trois principales catégories qui seront détaillées dans les sections suivantes : les claviers matériels modifiés (section 2.3.1), les logiciels de simulation de clavier par émulation d'un clavier à l'écran (section 2.3.2), et les logiciels de simulation de clavier qui améliorent la vitesse de saisie des utilisateurs par un système de prédiction des mots (section 2.3.2).

2.3.1 Les claviers matériels modifiés

Au vue de la rapidité de frappe de certaines personnes sur leur clavier d'ordinateur AZERTY ou QWERTY, il est logique de penser que l'organisation des touches de ces claviers a été mise au point dans le but de maximiser la vitesse de saisie. Or il n'en est rien. L'organisation actuelle des touches des claviers n'est en effet optimisée ni pour le confort, ni pour une saisie rapide. La disposition QWERTY a été mise au point pour les machines à écrire par Christopher Latham Sholes en 1868 et est devenue populaire en 1873 lors de leur production massive [182]. Cette organisation des touches a été établie pour que les lettres fréquemment contiguës dans les mots de la langue anglaise se trouvent éloignées sur le clavier, ceci afin de minimiser les risques d'entrechoquement des marteaux des machines à écrire. Cette disposition a donc été mise en place pour ralentir la vitesse de saisie, à cause de contraintes matérielles. Le clavier QWERTY des pays anglo-saxons a été réorganisé pour ainsi donner le clavier AZERTY pour les pays francophones ou encore le clavier QWERTZ en Allemagne.

Ce problème de marteaux est aujourd'hui dépassé. Ainsi, des recherches ont été effectuées pour trouver l'organisation optimale des touches d'un clavier afin de maximiser la vitesse de saisie. August Dvorak et William Dealey ont créé le clavier Dvorak [180] en 1936, avec une disposition des touches optimisée pour le confort et la vitesse de saisie des textes anglo-saxons. Pour le réaliser, ils ont essentiellement respecté quatre contraintes : (1) maximiser l'alternance des mains lors de la frappe ; (2) répartir les touches de façon équitable sur

les deux mains (ambidextrie) ; (3) utiliser en priorité la rangée centrale ; (4) en cas de non alternance des mains sur deux lettres consécutives, diriger la frappe vers le centre du clavier pour solliciter davantage les doigts les plus habiles (index et majeur). La figure 2.13 représente l'organisation des touches de ce clavier.



Fig. 2.13 – Organisation des touches du clavier Dvorak.

La version anglaise du clavier Dvorak a été standardisée par l'ANSI (American National Standards Institute) en 1982. Il n'existe pas de version française normalisée de ce clavier, ce qui a conduit à plusieurs propositions [216] [146] qui diffèrent par la position de quelques touches périphériques. Bien que présentant une plus grande ergonomie que les claviers QWERTY et AZERTY, ce clavier ne fut jamais adopté car la plupart des personnes l'ayant testé et ayant une grande aisance avec le clavier AZERTY (ou QWERTY) ont refusé de réapprendre un nouveau clavier [215].

Les claviers ergonomiques

Les essais de réorganisation des touches pour améliorer la vitesse de saisie sont peu adoptés par les personnes habituées au clavier traditionnel. Néanmoins, les utilisateurs se dirigent progressivement vers des claviers dont les touches sont positionnées différemment pour améliorer le confort. On parle de claviers ergonomiques. Des études concernant l'impact du design du clavier sur le confort et les performances ont prouvé que la position des mains, des poignets et des bras est très importante pour le confort d'utilisation du clavier. Aussi convient-il d'utiliser un clavier qui ne soit pas trop incliné vers l'arrière — comme c'est souvent le cas des claviers standards — pour ne pas casser le poignet. Pour la même raison, le clavier doit être le moins épais possible. Il est également recommandé d'utiliser un clavier possédant deux regroupements distincts de touches, à savoir un pour chaque main. L'espacement et l'angle entre ces deux parties permettent en effet de conserver la position naturelle des mains en prolongement des avant-bras. Les études de comparaison entre claviers standards et ergonomiques sur de courtes périodes d'utilisation (1h [53] et 2h [64]) n'ont pas donné de résultats en faveur des claviers ergonomiques ; en revanche, après un temps d'adaptation au clavier ergonomique [128], il a été remarqué des améliorations du confort, de la posture de l'utilisateur et de ses performances en terme de vitesse.

Précisons que les claviers ergonomiques se divisent principalement en trois catégories : les claviers fixes, les claviers réglables et les claviers sculptés.

Tout d'abord, les claviers fixes ergonomiques sont les plus populaires (figure 2.14) car ils diffèrent peu des claviers standards. L'organisation des touches est en effet identique, mais ils possèdent en plus un repose-poignet à l'avant du clavier ainsi que la possibilité de surélever l'arrière du clavier. Toutefois on observe des différences comme la dimension (ils sont légèrement plus longs que les claviers standards d'environ 5 cm) et la disposition des touches en deux zones inclinées et séparées par un angle d'ouverture.



Fig. 2.14 – Clavier ergonomique fixe (Logitech).

Ensuite, les claviers réglables présentent deux sections de touches complètement amovibles avec un angle d'ouverture et d'inclinaison réglable entre les sections (figure 2.15). Le point fort de ce type de clavier est sa totale adaptation à l'utilisateur. Reste que ces types de claviers sont plus difficiles à trouver en magasin et sont plutôt réservés à des personnes capables de saisir du texte « en aveugle » car l'inclinaison trop forte des deux sections de touches ne permet plus à l'utilisateur de voir les touches. Le « maxim keyboard » (figure 2.15 (a)) de la société Kinesis Ergo [211] permet de choisir l'angle de séparation (de 0° à 30°) des deux parties du clavier pour un meilleur confort d'utilisation. Les supports de poignets sont démontables comme sur la plupart des claviers traditionnels. Le clavier « evolution » (figure 2.15 (c)) se compose de deux parties distinctes que l'utilisateur peut placer comme il le désire. Une déclinaison de ce modèle comporte un support permettant de régler l'inclinaison des deux parties, ou encore la distance entre les deux parties du clavier (figure 2.15 (b)).



(a) (b) (c)

Fig. 2.15 – Claviers réglables (Kinesis Ergo) : maxim keyboard (a), freestyle keyboard (b) et evolution keyboard (c).

Enfin, les claviers sculptés (figure 2.16) possèdent d'une part les caractéristiques habituelles des claviers ergonomiques, à savoir la séparation en deux parties de la zone alpha-

numérique et l'angulation pour favoriser une position neutre des poignets. D'autre part, ces claviers séparent très nettement les deux zones de touches qui sont placées de manière concave. Cette disposition permet de tenir compte de la longueur des doigts. Dans ce type de clavier, la disposition des touches diffère de celle des claviers standards mais également l'organisation des touches périphériques, ce qui implique un temps d'apprentissage de la part de l'utilisateur. A cela s'ajoute un inconvénient dut à la nature même du clavier : son épaisseur.



(a) (b)

FIG. 2.16 – Clavier sculptés : coutoured keyboard (Kinesis Ergo).

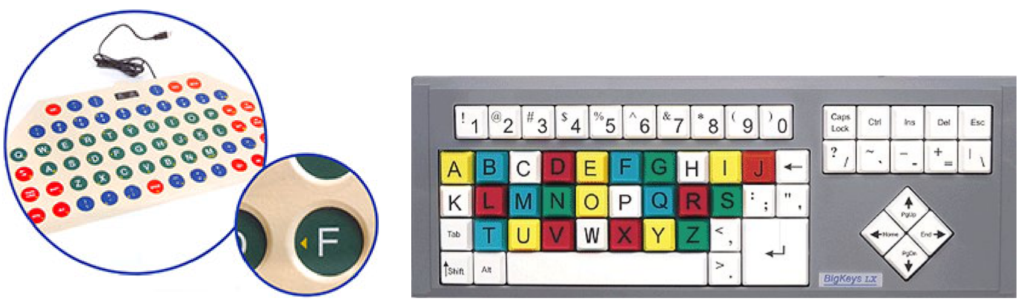
Des études ont été menées en reprenant les principes de claviers ergonomiques et d'amélioration de la vitesse de saisie par réorganisation des touches. Eggers and al. [56] ont travaillé à optimiser l'organisation des touches sur un clavier ergonomique (avec séparation des touches en deux parties) pour une langue donnée. Ils ont évalué la qualité de leur clavier en se basant sur des critères ergonomiques visant à améliorer le confort et la vitesse. Se basant sur un texte de référence pour générer l'organisation des touches du clavier, ils ont cherché à minimiser le nombre de lettres fréquemment utilisées et positionnées à des emplacements sur le clavier prévus pour être frappés par des doigts peu habiles comme l'auriculaire. Un autre de leurs objectifs est de placer deux lettres fréquemment successives (dans le texte de référence) dans les deux zones afin de minimiser les touches successives à frapper d'une seule main. Et lorsque cela n'est pas possible, ils ont minimisé le nombre de lettres successives à frapper par le même doigt. En se basant sur ces critères ergonomiques et sur la fréquence des successions de lettres dans une langue, ils ont généré un clavier français, un allemand et un anglais. Ils ont ensuite montré que les claviers qu'ils ont généré présentent de meilleurs résultats que le clavier AZERTY (et QWERTY), ou le clavier Dvorak. Précisons que pour trouver l'organisation des touches sur le clavier, ils ont utilisé un algorithme de fourmis artificielles qui sera présenté dans la section 3.4.3.

Les claviers alternatifs

Les claviers ergonomiques sont utilisés par tout type de personne simplement pour améliorer leur vitesse de frappe et leur confort. Sur de tels claviers, l'organisation des touches principales entre elles reste identique à celle des claviers traditionnels ; seule la disposition

des touches est modifiée. Il en existe également d'autres qui permettent de s'adapter plus particulièrement aux besoins des personnes handicapées et dont certains modifient l'organisation des touches.

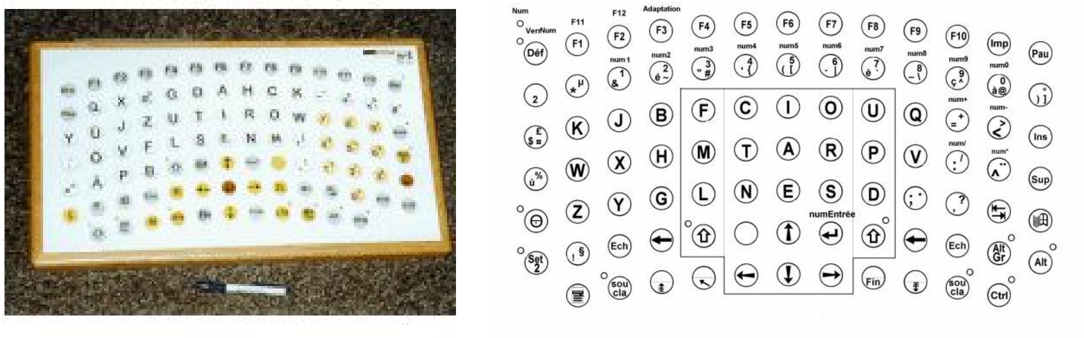
Pour commencer, on peut penser aux claviers grand format. Les touches sont plus larges que sur un clavier conventionnel, ce qui permet une utilisation plus facile pour les personnes malvoyantes ou les personnes ayant des tremblements. Pour ces dernières, les grandes touches espacées les unes des autres évitent d'en saisir plusieurs en même temps. Le « King keyboard » de la société Tash [242] possède une dimension extérieure de 54 ∗ 28 cm (tandis qu'un clavier conventionnel mesure environ 45 ∗ 15 cm). Ses touches rondes sont bien espacées et ont un diamètre de 3,1 cm (contre 1,5 ∗ 1,5 sur un clavier standard). Il est disponible en version QWERTY (figure 2.17 (a)) ou selon la fréquence des lettres en anglais — les lettres les plus souvent utilisées sont alors disposées sur la rangée centrale. Le clavier SUMO (figure 2.18 (a)) ne propose que ce deuxième mode d'organisation des touches. La figure 2.18 (b) présente son organisation des touches en version française. Il mesure 55 ∗ 32 cm et possède des touches d'un diamètre de 2,4 cm en creux espacées de 3,5 cm. Quant au clavier BigKeys [167] (figure 2.17 (b)), il propose l'organisation QWERTY mais également une organisation des touches par ordre alphabétique. Il est dit simplifié puisqu'il ne possède que 60 touches (de 2,5cm de côté). La version colorée par ordre alphabétique est destinée plus particulièrement aux enfants, tandis que les adultes et les non-voyants pourront opter pour la version QWERTY en noir et blanc.



(a) (b)

Fig. 2.17 – Claviers grand format : King keyboard de Tash(a) et BigKey (b).

De plus, le clavier BigKeys peut être utilisé avec un guide-doigts (figure 2.19 (a)). Les guide-doigts peuvent être ajoutés à tous les types de claviers (figure 2.19 (b)) et constituent de ce fait une aide technique à part entière. Ils sont destinés à faciliter l'utilisation des claviers pour les personnes ayant des difficultés motrices. Les guides facilitent la discrimination visuelle des touches, la visée et le maintien du doigt sur une touche. Ils permettent notamment d'éviter de frapper plusieurs touches avec un seul doigt et sont particulièrement indiqués pour les personnes souffrant de tremblements, d'imprécision du geste et éprouvant des difficultés à visualiser et viser les touches d'un clavier. Nous retrouvons ici les mêmes types d'utilisateurs que pour les claviers grand format. BigKeys propose un guide-doigts en option et les claviers King et SUMO en sont équipés nativement. S'il n'existe pas de guide-doigts pour un clavier, il est possible de le faire réaliser sur mesure [198].



(a) (b)

Fig. 2.18 – Clavier grand format SUMO de SEVEKE (a) et son organisation des touches en version française (b).



(a) (b)

Fig. 2.19 – Guide-doigt : en option sur le clavier BigKey (a), version en aluminium pour ordinateur portable (b).

A l'opposé, il existe des claviers de taille réduite, qui permettent de réduire les déplacements de l'utilisateur. Ils sont utilisés par les personnes ayant des difficultés motrices, comme la fatigue musculaire, un rayon d'action réduit ou encore par des personnes n'ayant l'usage que d'une seule main. La figure 2.20 (a) représente le mini clavier Datalux qui ne mesure que $27 * 15$ cm. D'autres claviers, comme le M42 (figure 2.20 (b)) et le M62 de Seveke, sont encore plus petits ($20 * 11$ cm). La taille des touches est également plus petite (environ 1 cm de côté) et il est alors possible de les actionner à l'aide d'un bâtonnet. Notons que ces deux claviers sont quasiment identiques et ne se distinguent que par la distance entre les touches légèrement supérieure sur le M62. Il existe également des claviers encore plus petits utilisables avec une tige à bouche. Ceux-ci ont la particularité de pouvoir être utilisés même avec une faible pression exercée sur les touches, et aussi sans aucun effort d'appui grâce à un bâtonnet à embout magnétique (figure 2.20 (c)). Dans ce type de clavier, les touches les

plus utilisées sont placées au centre afin de minimiser les mouvements de la tête.



(a) (b) (c)

FIG. 2.20 – Mini-claviers : Datalux (a), M42 de SEVEKE (b) et clavier GT de GORLO (c).

On peut également trouver des claviers ergonomiques permettant de faciliter la saisie à une main. Ces claviers « une main » ont une forme et une disposition des lettres spéciales en fonction de la fréquence d'apparition des lettres. La figure 2.21 (a) représente le clavier ergonomique MALTRON [218] pour main droite. A l'instar des claviers ergonomiques sculptés, il possède une forme creuse pour adapter la hauteur des touches à la longueur des doigts.



FIG. 2.21 – Clavier ergonomique MALTRON pour main droite.

2.3.2 Les claviers virtuels

Certains handicaps ou maladies peuvent entraver l'utilisation d'un clavier traditionnel, voire d'un clavier matériel adapté. C'est le cas notamment des personnes qui ne peuvent utiliser leurs bras et leurs poignets. Il est alors possible de substituer au clavier matériel un clavier virtuel, encore appelé clavier simulé ou clavier à l'écran. Grâce à ce type de clavier, l'utilisateur se sert de ses doigts pour valider les touches sélectionnées (souvent en utilisant un track-ball à la place d'une souris). Une étude menée par N. Vigouroux et *al.* [134] a permis d'analyser le comportement des internautes handicapés moteurs face à un questionnaire en ligne — ce dernier servant d'ailleurs pour recueillir les informations données par l'utilisateur

sur les problèmes qu'il rencontre sur Internet. Il a été soulevé que le temps de saisie est un élément négatif lors des échanges sur Internet. De nombreuses approches ont été envisagées et seront présentées par la suite, pour améliorer le confort et la vitesse de saisie grâce aux claviers virtuels, telles la redisposition ou la réorganisation des touches, ou encore l'ajout d'un système de prédiction pour les claviers virtuels. Remarquons de plus que les claviers virtuels sont de plus en plus utilisés, pour la saisie avec un stylet, dans les technologies nomades telles les téléphones portables, les ordinateurs de poches (PDA), les navigateurs GPS...

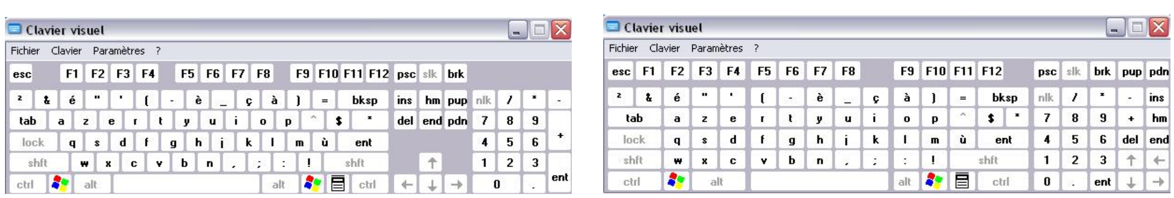
Les attentes face à ce type de claviers sont différentes de celles des claviers matériels. Il s'agit ici d'optimiser la saisie à « un doigt » (et non à 10, ou 5), que ce soit avec une souris, un trackball, un stylet... Les déplacements pour atteindre les touches sont beaucoup plus nombreux que pour un utilisateur averti de clavier matériel utilisant ses dix doigts. Deux objectifs sont à envisager : l'accélération de la saisie et l'amélioration du confort (ou, pour les personnes handicapées, la facilité d'utilisation). Pour cela, plusieurs pistes sont étudiées, comme la manière de naviguer sur un clavier virtuel pour atteindre une touche, l'utilisation d'un système de prédiction ou encore la réorganisation des touches du clavier.

La sélection des touches

Pour saisir une touche, il faut déplacer le curseur puis valider. Les personnes capables de réaliser ces deux actions, même si elles induisent des fatigues musculaires et oculaires, peuvent utiliser un clavier virtuel tel celui fourni en standard sous Windows. Le projet ChatCom [102] a comparé les performances de personnes myopathes sur trois types de claviers virtuels (windows, avec technologie bigkeys au survol de la souris, et sans pointeur visuel). Des résultats similaires, concernant la vitesse de saisie et la rapidité d'accès à une touche, ont été obtenus pour les deux premiers types de clavier, alors que le dernier clavier a donné de moins bons résultats (probablement à cause de la fatigue visuelle engendrée par l'absence de pointeur à l'écran). Ces résultats étant à relativiser selon les personnes, qui dans cette étude étaient tous des utilisateurs du clavier de windows. Toutefois, les actions « déplacer » et « valider » ne sont pas réalisables par toutes les personnes handicapées même si elles utilisent l'interface physique la plus adaptée : souris, joystick, trackball... Il faut savoir que certaines personnes handicapées sont incapables d'utiliser totalement ces dispositifs de pointage : certaines ont uniquement la capacité de déplacer le pointeur mais pas de valider en cliquant sur le bouton, alors qu'il peut se produire l'inverse chez d'autres personnes. Les claviers virtuels doivent alors s'adapter à tous les cas de figure en proposant des alternatives à l'interface matérielle de pointage, que ce soit lors de la sélection ou lors de la validation.

Certains claviers, comme par exemple le clavier virtuel de Windows XP, le clavier Click-N-Type [214], le clavier CVK [206], le clavier ScreenDoors 2000 [217] ou encore le clavier Keystrokes [163] pour Macintosh, proposent l'utilisation de la fonctionnalité « autoclic » encore appelée « clic temporisé ». Cette option valide automatiquement la touche sélectionnée si le curseur reste immobile. La durée de la temporisation avant validation est paramétrable.

Pour les personnes ne pouvant sélectionner la touche uniquement à l'aide du dispositif de pointage, des modes de balayage automatique existent : le défilement linéaire, le défilement bi-directionnel ligne/colonne, et le défilement par bloc. Le défilement linéaire permet de voir les lettres défiler une par une, ligne par ligne. L'utilisateur doit valider la touche lorsqu'elle est sélectionnée. Le défilement bi-directionnel permet quant à lui de sélectionner dans un premier temps la ligne où se trouve la touche, puis, dans un second temps la touche en question. Ce type de balayage est plus rapide que le simple défilement linéaire — et est de ce fait le mode de balayage proposé par le clavier windows — mais nécessite plus de validations de la part de l'utilisateur. Afin d'accélérer ces modes de défilement, certains claviers proposent un défilement par bloc. Couplé au défilement linéaire, il permet de choisir le bloc où va s'effectuer le balayage des touches ; et couplé au défilement bi-directionnel, la ligne puis le bloc devront être validés avant d'atteindre la touche. La figure 2.22 (a) illustre le clavier virtuel de Windows XP dans sa configuration standard. L'organisation des touches reste la même que sur les claviers matériels. Lors d'un défilement linéaire, les premières touches accessibles sont les touches de fonction, et la barre d'espace — très utilisée — n'est accessible qu'après être passé sur toutes les touches des 5 lignes précédentes. Par exemple, pour atteindre la lettre « k » en défilement bi-directionnel, l'utilisateur doit d'abord attendre 4 défilements et valider pour atteindre la 4ème ligne, puis attendre 9 défilements avant de valider la lettre « k », soit au total 13 défilements et 2 validations. Afin d'accélérer le défilement, ce clavier offre une deuxième organisation par bloc (figure 2.22 (b)) que l'on trouve également sur le clavier Clavicom [200]. Reprenons notre exemple de saisie de la lettre « k » : les 4 premiers défilements et la première validation permettent d'atteindre la 4ème ligne, puis le bloc contenant la lettre « k » est atteint après 2 défilements et une validation, enfin la lettre est atteinte dans ce bloc après 5 défilements et une validation. Au final, il aura fallu dans ce mode 11 défilements et 3 validations pour atteindre la lettre « k ». Ce mode de balayage par bloc permet ainsi d'accéder à une touche du clavier avec moins de pas de défilement (action nécessitant un temps paramétrable) mais plus de validations qu'avec le mode linéaire. Notons également que si le nombre de défilement est réduit pour accéder aux touches situées dans les blocs les plus à droite du clavier, l'accès aux touches du premier bloc verra son nombre de validations augmenter mais ne verra pas son nombre de défilement diminuer.



(a) (b)

Fig. 2.22 – Clavier virtuel de Microsoft Windows XP : disposition standard (a), et disposition par bloc (b) pour le défilement automatique.

L'utilité d'accélérer la saisie en utilisant des modes de défilement différents dépendra alors des possibilités de l'utilisateur concernant l'action de validation. Les personnes éprouvant des

difficultés à exécuter cette action ne verront pas d'avantages au défilement bi-directionnel ou par bloc et continueront d'utiliser le défilement linéaire qui ne requiert qu'une seule action de validation par lettre.

Les systèmes de prédiction

Pour optimiser la vitesse de saisie et réduire la fatigue, certains claviers virtuels utilisent un système de prédiction. Généralement, les mots les plus probables sont présentés sous forme de liste à l'utilisateur. Celle-ci correspond aux mots appartenant au dictionnaire du système et pouvant convenir à partir des premières lettres saisies. Il s'agit donc d'une étude linguistique hors contexte : les lettres du mot sont prises en compte mais pas le rôle de ce dernier dans la phrase. Les accords avec les mots précédemment tapés ne sont donc pas effectués. Remarquons que l'ordre de la liste générée peut être alphabétique, selon la fréquence d'utilisation ou encore selon les plus récentes utilisations. Les claviers virtuels cités précédemment (hormis le clavier virtuel de Windows XP) utilisent ce système de prédiction. Le clavier Vitipi [13] est même capable de s'adapter aux erreurs de frappe de l'utilisateur — Vitipi signifiant d'ailleurs « Version Interprétant un Texte Imparfaitement écrit pour les Personnes Inexpérimentées ». Face à un mot n'existant pas dans son dictionnaire, le système a la possibilité, pour obtenir un mot qui appartient à son lexique, de : (1) substituer une lettre à une autre, (2) rajouter une lettre dans un mot, et (3) supprimer une lettre du mot. Ajoutons que dans le premier cas la substitution sera réalisée uniquement si la lettre utilisée dans le mot et la lettre de substitution sont voisines sur le clavier (ceci pouvant expliquer la faute de frappe).

Le système de prédiction utilisant un dictionnaire peut également être couplé à une prédiction grammaticale. Dans cette dernière, le système propose des mots de la catégorie grammaticale la plus probable (nom, adjectif, verbe...) en fonction du contexte gauche. C'est le cas du clavier HandiAs [88] qui a été défini pour la grammaire française mais qui peut être adapté pour d'autres langues [89].

La figure 2.23 représente le clavier CVK et son système de prédiction de mots. Ce clavier est open source et peut s'utiliser dans les applications standards de Windows (ici le bloc notes).

Pour évaluer ce type de système, la qualité du dictionnaire est bien entendu déterminante. Un autre critère important des systèmes de prédiction concerne l'adaptation à l'utilisateur. Dans un premier temps, on peut trouver des claviers actualisant la fréquence associée aux mots selon ce que saisit l'utilisateur (ce qui influera sur l'ordre de la liste de mots proposés). C'est le cas du clavier CVK et de sa fonction de prédiction « auto-apprenante ». Dans un deuxième temps, il existe des claviers dont le dictionnaire peut s'enrichir de vocabulaire nouveau. Cet ajout dans le dictionnaire peut se faire à la demande de l'utilisateur (clavier Clavicom [200]), ou de manière automatique dès qu'un mot n'appartenant pas au dictionnaire est saisi (clavier Screendoors [217]). Si le mode automatique peut être intéressant pour les personnes handicapées, il ne faut pas oublier qu'il contribuera à la « pollution » du dictionnaire en y ajoutant bon nombre de mots mal orthographiés par l'utilisateur.

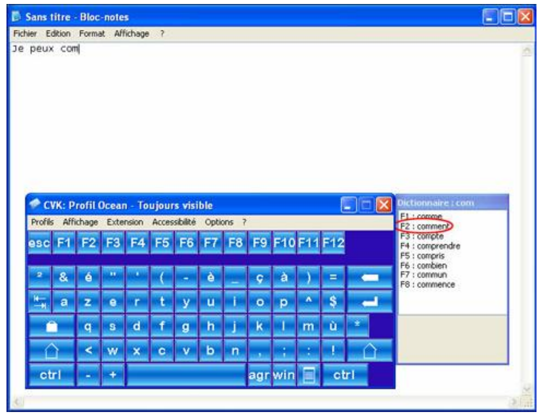


FIG. 2.23 – Système de prédiction de mots du clavier CVK.

Hormis les systèmes de prédiction de mots, on peut également trouver des claviers munis d'un système avec prédiction de lettres. C'est le cas des claviers Keyglasses [109] et Sibylle [105]. Cette méthode de prédiction de lettres est fondée sur le modèle statistique n-gramme, permettant d'estimer la probabilité d'une lettre en fonction des $(n-1)$ lettres précédentes. Avec le clavier Keyglasses, les lettres qui ont la plus forte probabilité d'être saisies apparaissent dynamiquement en semi-transparence autour de la dernière lettre saisie. La figure 2.24 représente ce clavier où la lettre « b » vient d'être saisie et où 4 lettres sont proposées autour de cette dernière. Après avoir cliqué sur une Keyglass, d'autres Keyglasses apparaissent à ses côtés. Ce clavier permet de réduire les déplacements pour saisir un texte ; par contre la vitesse se trouve quelque peu réduite [108]. Avec le clavier Sibylle, après chaque saisie, l'ordre des lettres est réorganisé de manière à présenter en priorité celles ayant la plus forte probabilité d'apparition [119][120]. L'objectif de ce clavier est d'améliorer la vitesse de saisie avec un défilement linéaire — qui nécessite moins de validation que le balayage bi-linéaire. Néanmoins, pour les personnes utilisant un autre mode de navigation, l'utilisation de ce clavier est difficile puisque l'organisation des touches sur le clavier change après chaque saisie de lettre.

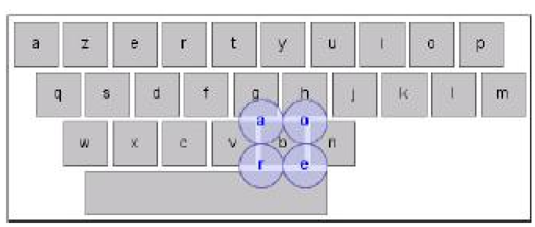


FIG. 2.24 – Clavier virtuel Keyglasses avec son système de prédiction de lettres : proposition de 4 lettres après la saisie de la lettre « b ».

Le clavier Vitipi utilise un système de prédiction à la frontière entre la prédiction de mots et celle de lettres. Les auteurs de ce clavier ont montré que la liste de mots proposés peut

perturber l'utilisateur [12]. Ils suggèrent alors d'insérer automatiquement la ou les lettres dans le mot (pouvant aller jusqu'à compléter totalement la saisie du mot) dès lors qu'il n'y a plus d'ambiguïté. Pour cela, le clavier Vitipi utilise un dictionnaire qui peut être complété par l'ajout de nouveaux mots.

La réorganisation des touches

A l'instar des claviers matériels, la réorganisation des touches du clavier virtuel peut améliorer la vitesse de saisie. Les claviers virtuels représentant des claviers QWERTY ou AZERTY ne sont pas optimisés pour la saisie avec un stylet ou avec une sélection par balayage (pas plus qu'ils ne sont optimisés pour la saisie à 10 doigts). Le clavier Dvorak possède une disposition des touches qui a été étudiée pour optimiser la vitesse de saisie à deux mains et non avec un pointeur. Pour trouver une organisation optimale pour clavier virtuel, la première solution envisagée concerne la fréquence d'apparition des lettres. L'alphabet ESARIN représente les lettres par ordre décroissant des fréquences d'apparition en français : E S A R I N T U L O M D P C E F B V H G J Q Z Y X K W.

L'utilisation d'un tel ordre des touches sur un clavier virtuel permet de favoriser l'accès aux lettres les plus fréquemment utilisées — au moins lors de l'utilisation du balayage linéaire. Toutefois, cette organisation peut être améliorée en utilisant le contexte de la langue française. La lettre « E », bien que la plus fréquente de la langue française, est rarement utilisée en début de mot. Il peut alors être judicieux d'y avoir accès rapidement lorsqu'il y a une forte probabilité de cette lettre soit la prochaine, et il s'agit alors d'un système de prédiction de lettre.

D'autres essais de claviers virtuels utilisant une réorganisation des touches ont été réalisés. Lesher et Moulton [79] ont étudié la possibilité de réaliser un clavier virtuel pour les personnes handicapées. Dans leur méthode, à chaque transition entre deux touches est affecté un coût qui dépend de la distance entre les centres de celles-ci. L'efficacité du clavier est évaluée en sommant les coûts des transitions entre les touches, en se basant sur les fréquences de co-occurence des lettres. Des permutations de couples de lettres sont ensuite réalisées avant de réévaluer le clavier et de le conserver s'il est meilleur que le précédent (algorithme n-opt [81]). Ils ont montré une réduction du nombre de déplacements de 32% lors de l'utilisation du clavier KNITS (configuration $5 * 6$ illustrée par la figure 2.25) par rapport au clavier alphabétique, et une réduction de 35% avec la configuration $4 * 10$ (figure 2.26) par rapport au clavier QWERTY.

	0	1	2	3	4	5
0	Z	G	C	H	V	X
1	K	N	I	T	S	W
2	Espace		A	E	Espace	
3	Y	D	L	R	O	F
4	Q	B	U	M	P	J

FIG. 2.25 – Clavier KNITS au format 5×6.

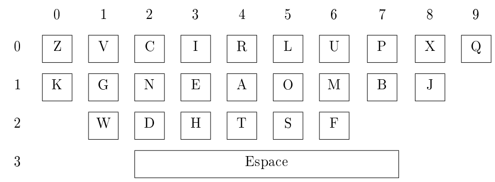


FIG. 2.26 – Clavier « KNITS » au format 4×10.

Le clavier Métropolis [144] est destiné à la saisie à l'aide d'un stylet (pour un PDA) avec l'objectif de minimiser la longueur des déplacements entre les touches. La fonction coût utilisée pour évaluer la qualité du clavier est la loi de Fitts [57][58] qui précise que le temps de déplacement pour aller d'une touche à une autre dépend de la taille et de la distance entre les touches. Aussi, le temps pour aller de la touche i à la touche j d'après la loi de Fitts est donné par l'équation suivante,

$$T_{ij} = a + bLog_2(D_{ij}/(W_j + 1)) \tag{2.1}$$

où D_{ij} représente la distance entre les touches i et j, W_j la largeur de la touche j, et a et b deux constantes qui dépendent des capacités de l'utilisateur. Il est de ce fait possible d'estimer le temps moyen t_j de saisie de la lettre j en utilisant P_{ij} qui correspond à la probabilité de saisir la touche j après la touche i :

$$t_j = \sum_{i=1}^{n} P_{ij} * T_{ij} \tag{2.2}$$

avec n le nombre de touches dans le clavier. L'algorithme développé pour ce clavier vise à minimiser le temps moyen pour atteindre chaque touche du clavier soit $\sum_{j=1}^{n} t_j$. La figure 2.27 illustre l'organisation des touches du clavier Métropolis. Nous constatons que les résultats ont permis de montrer un gain de 43% au niveau du nombre de mots par minute (wpm) par rapport au clavier QWERTY.

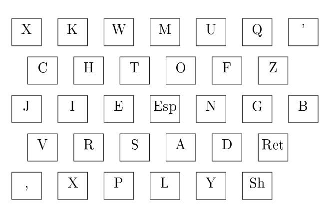


FIG. 2.27 – Clavier Métropolis.

D'autres études [82][83] montrent l'impact de la configuration spatiale des touches sur la vitesse de saisie. Les résultats sont synthétisés dans le tableau 2.2.

Claviers virtuels	Vitesse de saisie (en wpm)	
	utilisateur novice	utilisateur expérimenté
organisation Dvorak	8,7	38,7
organisation alphabétique	9,6	40,9
organisation Qwerty	8,9	43,2
organisation Fitaly	9,7	55,9
organisation OPTI	17	60

TAB. 2.2 – Evaluation de la vitesse de saisie de divers claviers virtuels.

Nous observons sans étonnement que les plus mauvaises performances sont obtenues par l'organisation Dvorak. Remarquons également que l'organisation alphabétique a donné de meilleurs résultats que l'organisation QWERTY pour les débutants, qui ont pu repérer rapidement les lettres sur cette organisation bien connue. Les claviers Fitaly [243] (développé pour la saisie sur PDA) et OPTI [145] ont été conçus pour minimiser les déplacements d'une touche à une autre. Leur spécificité réside dans la disponibilité de plusieurs touches « Espace » (deux pour Fitaly et quatre pour OPTI), placées au centre pour être facilement accessibles.

La plateforme de conception SOKETO [133] permet à l'utilisateur de créer lui-même son propre clavier virtuel (taille, couleur, espacement et organisation des touches). La comparaison de deux claviers générés par ce biais, face au clavier virtuel AZERTY a permis là encore de montrer l'inadaptation de l'organisation AZERTY par rapport à une autre organisation des touches [132]. Notons que dans cette étude, les claviers ont été générés par deux personnes myopathes, auxquelles il a été demandé de réaliser un clavier visant à réduire leurs propres difficultés motrices lors de la saisie.

Par conséquent, le choix de la configuration spatiale des touches est primordiale pour accélérer la vitesse de saisie et diminuer la fatigue musculaire. Toutefois, à l'instar du clavier Dvorak, le problème de l'apprentissage d'une nouvelle organisation des touches du clavier se pose. Des études [82][83] ont en effet montré qu'un temps d'adaptation est nécessaire avant que l'utilisateur n'obtienne de meilleures performances avec le clavier OPTI plutôt qu'avec le clavier virtuel QWERTY.

2.4 Conclusion

Nous avons, dans la première partie de ce chapitre, envisagé une solution au problème de navigation au sein d'un site web qui consiste en l'utilisation d'un plan du site. Ce dernier devrait, d'après les recommandations de la WAI, être présent sur tous les sites web. En attendant la mise en application de ces directives par les webmestres, les internautes peuvent utiliser un générateur de plan. Toutefois, bon nombre de ces générateurs sont destinés aux webmestres plutôt qu'aux internautes. Nous nous intéresserons dans le chapitre 5 aux générateurs de plan en en proposant un qui soit adapté plus particulièrement aux internautes déficients visuels.

Dans la deuxième partie de ce chapitre, nous avons remarqué que certaines personnes à mobilité réduite peuvent avoir des difficultés à saisir du texte, et donc à profiter pleinement des services offerts par Internet. Pour les aider lors de la saisie de texte, nous avons présenté divers types de claviers, qu'ils soient matériels ou bien virtuels. Dans les deux cas, le but est de proposer une alternative au clavier traditionnel inadapté. Nous nous focaliserons par la suite (chapitre 4) plus particulièrement aux claviers virtuels avec réorganisation des touches, qui rappelons-le, ont montré des résultats concluants malgré un temps d'adaptation nécessaire.

Chapitre 3

Les fourmis artificielles

Ce chapitre aborde une heuristique trouvant son inspiration dans le comportement des fourmis réelles qui peut être transposé en comportement artificiel pour résoudre des problèmes complexes, comme ceux d'optimisation combinatoire. Nous aborderons ici les premiers algorithmes de résolution du problème du voyageur de commerce pour finir par la résolution d'autres méthodes liées aux réseaux par des fourmis artificielles.

3.1 Introduction

La biologie a toujours été une source d'inspiration pour les chercheurs. C'est ainsi que les réseaux de neurones et les algorithmes génétiques, qui sont des approches largement étudiées et reconnues, s'inspirent du vivant. Plus récemment, les scientifiques se sont tournés vers les insectes sociaux capables, grâce à leur comportement collectif évolué, de résoudre des problèmes complexes. Les fourmis, qui font partie de ce type d'insectes, seront présentées dans ce chapitre. Dans la réalité, les fourmis sont auto-organisées et aptes à faire face à de nombreuses difficultés qu'une unique fourmi ne pourrait résoudre à elle seule. L'organisation de ces insectes ainsi que leur mode de communication sont étudiés dans la section 3.2. Ce domaine de recherche qu'est l'intelligence en essaim permet de transposer les capacités de ces insectes en techniques artificielles de résolution de problèmes. La section 3.3 présente les premières applications historiques des algorithmes de fourmis artificielles au problème du voyageur de commerce. Pour finir, la section 3.4 s'intéresse aux algorithmes de fourmis artificielles pour d'autres problèmes d'optimisation combinatoire, qui nous permettrons par la site de résoudre des problèmes liés au handicap.

3.2 Les fourmis réelles

3.2.1 Généralités

Les fourmis sont des insectes présents dans des environnements très variés comprenant les milieux terrestre (forêts vierges, déserts. . .) et souterrain. Cela s'explique par le nombre important d'espèces (plus de 10 000 connues) dont chacune dispose d'une morphologie et d'un comportement particulier pour s'adapter à son milieu [68]. Elles vivent en colonie dans un nid appelé fourmilière qui peut là encore prendre diverses formes selon les espèces. Par exemple, les fourmis tisserandes (*oecophylla*) construisent leur nid dans les arbres en utilisant les feuilles alors que les fourmis rousses (*formica*) fabriquent leur nid sur la terre. Au cours de leur développement, les fourmis passent par quatre étapes successives : œuf, larve, nymphe, puis le stade adulte. A savoir, les larves n'ont pas de pattes et sont dépendantes des autres fourmis adultes qui doivent s'occuper d'elles. Quelques semaines plus tard les larves deviennent des nymphes qui sont nues ou dans un cocon sécrété à la fin du stade larvaire selon les espèces. A la fin du stade nymphal, la fourmi est alors adulte et peut commencer à travailler.

3.2.2 Les différentes tâches réalisées par les fourmis

Individuellement les fourmis ont un comportement simple ; elles savent par exemple mémoriser l'environnement proche du nid. Pour faire face à leurs capacités individuelles minimes, elles ont développé un comportement commun complexe. Les fourmis vivent effectivement en colonie et font preuve d'une grande intelligence collective (intelligence en essaim). C'est pourquoi nous parlons d'insectes sociaux. Les fourmis se regroupent en colonie, laquelle est auto-organisée. Il y a bien la présence d'une reine dans la fourmilière mais celle-ci n'a aucun pouvoir sur ses congénères. Les fourmis doivent alors s'organiser collectivement pour réaliser diverses tâches :

- nourrir, nettoyer et protéger le couvain ;
- construire et entretenir le nid ;
- rechercher de la nourriture (fourragement) ;
- défendre le nid.

Toutes ses tâches doivent être accomplies simultanément. Les fourmis se partagent pour cela le travail en se spécialisant pour une tâche en particulier.

3.2.3 Organisation des fourmis en castes

Dans la fourmilière, trois types d'individus cohabitent : les femelles fertiles, les femelles stériles et les mâles. Les fourmis ayant des aspects et des rôles différents, elles sont regroupées en castes dans la fourmilière :

- la caste des reproducteurs, qui comprend les mâles et les femelles fertiles (les futures reines) ;

- la caste des ouvrières, c'est-à-dire les femelles stériles, qui assurent toutes les tâches dans la fourmilière.

Les reproducteurs mâles possèdent des ailes et sont inactifs : ils vivent dans le fond de la fourmilière jusqu'au vol nuptial où ils féconderont les femelles fertiles. Peu après l'accouplement, les mâles meurent. Les femelles fécondées peuvent alors devenir des reines. Chez certaines espèces de fourmis, la reine est incapable de créer sa propre colonie — c'est le cas de la fourmi rousse. Elle doit alors usurper la place de la reine d'une autre colonie ou bien se faire accepter dans une colonie où il y a plusieurs reines. A l'opposé, chez les fourmis *Atta* vivant en Amérique du Sud, la reine creuse elle-même une chambre souterraine pour y pondre ses œufs. N'ayant personne à son service pour la nourrir, elle mange ses ailes afin d'avoir l'énergie nécessaire pour pondre ses premiers œufs. Par la suite, tant qu'aucune fourmi ne sera adulte, la reine se nourrira d'une partie du couvain et de quelques-uns de ses œufs. A ce stade, la reine est alors polyvalente. Lorsque des fourmis auront atteint le stade adulte, elles s'occuperont elles-mêmes de tout dans la fourmilière, laissant ainsi à la reine la seule tâche de pondre.

Les ouvrières s'organisent elles aussi en caste :

- les nourrices ;
- les pourvoyeuses, comprenant les bâtisseuses et les fourragères ;
- les soldats/gardiens.

Chez certaines espèces, il s'agit de castes seulement temporelles qui sont alors définies comme une partie de la société spécialisée dans l'exécution d'une tâche pendant une certaine période de leur vie. Ainsi les plus jeunes ouvrières s'occupent de la reine et du couvain (nourrir, nettoyer, déplacer les larves dans les différentes chambres de la fourmilière pour qu'elles restent à température constante. . .), puis elles deviennent des bâtisseuses. Lorsqu'elles sont suffisamment âgées pour s'aventurer hors du nid, elles s'occupent de l'approvisionnement en nourriture puis de la défense du nid. En revanche, chez d'autres espèces de fourmis, les castes sont déterminées par des particularités morphologiques. Par exemple les larves destinées à être des soldats sont suralimentées. Les « ouvrières-soldats » sont plus grosses que les autres ouvrières, et possèdent d'énormes mandibules (habituellement utilisées par les ouvrières pour saisir et broyer la nourriture) en guise d'armes.

3.2.4 La communication

La communication entre les fourmis est très élaborée. Elles utilisent en effet plusieurs modes d'échange : acoustique, visuel, tactile, et chimique. Nous allons étudier plus en détail ce dernier mode de communication qui est celui le plus fréquemment utilisé par les fourmis. Cette communication s'effectue à l'aide de substances chimiques : les phéromones. Les fourmis possèdent diverses glandes capables de produire des phéromones ayant des objectifs différents :

- les glandes mandibulaires, pour alerter les autres fourmis d'un danger (message d'alerte) ;
- les glandes pygidiales pour faire fuir les fourmis étrangères (message d'avertissement) ;
- les glandes alcalines (ou de Dufour) pour tracer un rail odorant conduisant généralement à une source de nourriture (message de recrutement).

Ajoutons que les phéromones sont peu volatiles, ce qui permet au message chimique de subsister quelques temps avant de s'évaporer totalement. Par la suite nous nous intéresserons davantage au dernier type de phéromones.

Voici comment s'effectue l'action de recrutement avec les phéromones. Les ouvrières déposent des phéromones sur leur chemin entre le nid et une source de nourriture afin, d'une part, de retrouver leur chemin pour retourner au nid, et d'autre part dans le but de recruter d'autres fourmis. Les dépôts successifs de phéromones sur le terrain permettent de renforcer positivement certains chemins alors que l'évaporation naturelle des phéromones a une fonction d'oubli. Les chances de suivre un chemin fortement marqué en phéromones augmentent avec le nombre de passages des fourmis qui à leur tour renforcent le chemin. Ce mécanisme est illustré par la figure 3.1.

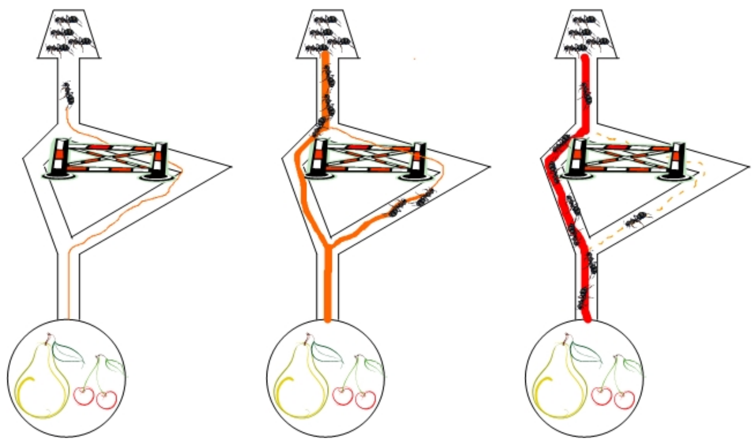


FIG. 3.1 – Exploitation d'une source de nourriture : l'éclaireuse découvre de la nourriture (a), les autres fourmis partent exploiter cette source de nourriture (b) et découvrent grâce à leurs phéromones le chemin le plus court (c).

Les fourmis se déplacent généralement en ligne droite mais lorsqu'un obstacle se présente, elles empruntent indifféremment un autre chemin. Sur la figure 3.1 (a) une fourmi (l'éclaireuse) a trouvé une source de nourriture et rentre au nid en déposant des phéromones sur le chemin. C'est grâce à leurs phéromones que les éclaireuses incitent leurs congénères à sortir du nid pour aller exploiter la source de nourriture découverte (phénomène de recrutement). Devant un obstacle éventuel, certaines fourmis vont emprunter le chemin le plus court et d'autres le chemin le plus long. Or, les fourmis ayant emprunté le chemin le plus court feront plus vite l'aller-retour (figure 3.1 (b)), ce qui implique que ce chemin sera d'autant plus renforcé par les phéromones. Ainsi, de plus en plus de fourmis seront incitées à prendre cette voie. Après un certain temps, les phéromones déposées sur le trajet le plus long, qui n'est plus emprunté, vont s'évaporer et seul sera marqué l'itinéraire le plus court (figure 3.1 (c)). Une conséquence particulière de ce mécanisme de communication est que les fourmis sont inconsciemment capables de trouver le plus court chemin entre le nid et une source de nourriture.

3.3 Naissance des fourmis artificielles

Nous savons que les fourmis possèdent des comportements très variés suivant les espèces et les environnements dans lesquels elles vivent. Les comportements développés par les fourmis pour résoudre leurs problèmes quotidiens (se nourrir, se déplacer, se reproduire…) sont, par leur approche collective et géographiquement dispersée, des sources d'inspiration pertinentes pour résoudre des problèmes informatiques, que nous souhaitons expérimenter sur des problèmes liés au handicap. Les fourmis sont des insectes sociaux qui utilisent différents signaux pour communiquer, la transmission chimique étant particulièrement utilisée. Aussi l'étude du comportement des fourmis a permis de montrer que le marquage par les phéromones des chemins entre le nid et une source de nourriture les incite à emprunter le plus court. En modélisant cette capacité des fourmis réelles, des méthodes de résolution de problèmes d'optimisation combinatoire ou de routage dans les réseaux peuvent être mises en place. La modélisation d'autres comportements des fourmis comme le tri du couvain peut également permettre de résoudre des problèmes de classification.

3.3.1 Les fourmis et le problème du voyageur de commerce

Les premiers travaux sur les fourmis artificielles ont consisté à appliquer la capacité des fourmis à trouver le plus court chemin, à la classique problématique combinatoire du voyageur de commerce [50][44] (nous utiliserons par la suite l'abréviation PVC, correspondant en anglais à TSP pour *Traveling Salesman Problem*).

Le PVC consiste, étant donné un ensemble de n villes séparées par des distances données, à trouver le plus court chemin qui relie toutes les villes, en n'y passant qu'une seule fois, et que ce chemin revienne au point de départ. Précisons que pour de grandes instances du PVC, les méthodes exactes ne permettent pas de trouver l'optimum en un temps raisonnable. Il a effectivement été démontré en 1979 que ce problème est NP-difficile [63]. Tester l'ensemble des solutions, même les plus absurdes, prendrait de ce fait trop de temps lorsque le nombre de ville est important, puisque l'espace de recherche correspond à l'ensemble des combinaisons possibles des n villes, soit $n!$ combinaisons. La seule solution consiste alors à utiliser des méthodes de résolution approchées — lorsqu'une solution approchée est acceptable. De nombreuses heuristiques ont été développées pour résoudre ce problème comme celles des plus proches voisins, des réseaux de neurones, des algorithmes génétiques ou encore des colonies de fourmis. Tel quel, ce problème peut s'appliquer à des situations de transport et de logistique (trouver le chemin le plus court pour les bus de ramassage scolaire, pour une tournée de poids lourd transportant de la marchandise…), mais il correspond plus généralement à la recherche du plus court chemin dans un graphe. De ce fait de nombreux problèmes de recherche opérationnelle se ramènent à celui du voyageur de commerce.

3.3.2 De Ant System à Ant Colony System

L'approche consistant à résoudre le PVC par des colonies de fourmis artificielles a été développée dans de nombreuses variantes comme par exemple avec l'algorithme Ant System

(AS) [51]. Dans cet algorithme (voir algorithme 1), m fourmis se déplacent entre les n villes en déposant sur les chemins entre ces villes des phéromones $\tau_{i,j}$ avec $(i,j) \in \{1..n\}$. Ce problème est modélisé en représentant chaque ville comme le sommet d'un graphe et les chemins reliant les villes comme les arêtes du graphe.

Algorithme 1 Ant System (AS) pour le PVC

```
Initialiser les valeurs des phéromones τ_ij
Répéter
   Pour tout fourmis f Faire
      Construire un chemin S^f
      Evaluer la longueur de ce chemin L^f
   Fin pour
   Mettre à jour la meilleure solution trouvée S*
   Mettre à jour la matrice de phéromones τ_ij = (1 − ρ)τ_ij + Σ_{f=1}^{m} Δτ_ij^f
Jusqu'à la validation du test d'arrêt
Retourner la meilleure solution S* trouvée depuis le début
```

A l'instar des fourmis réelles, lorsqu'une fourmi artificielle se déplace d'une ville i à une ville j, elle dépose des phéromones sur le chemin reliant ces deux villes. Précisons que des différences apparaissent entre les fourmis réelles et les fourmis artificielles de cet algorithme : ces dernières ne sont pas totalement aveugles, et ont de ce fait connaissance de la distance qui les sépare des autres villes ; elles possèdent une mémoire pour se souvenir des villes déjà visitées et ne pas y retourner.

Le choix de la direction que prend la fourmi est basé sur une probabilité qui dépend d'une part de la distance entre la ville où elle se trouve et les différentes villes qui lui reste à parcourir, et d'autre part de la quantité de phéromones déjà présente sur ces chemins. La probabilité qu'une fourmi choisisse de se diriger vers une certaine ville sera d'autant plus forte que la distance séparant ces villes sera petite et que la quantité de phéromones sera importante. Notons que grâce à la mémoire de la fourmi, la probabilité qu'elle choisisse de se diriger vers une ville qu'elle a déjà visitée est nulle. L'équation 3.1 donne cette probabilité. Précisons que cette probabilité est biaisée par les paramètres α et β qui déterminent l'importance de l'influence respectivement de la quantité de phéromones τ_{ij} et de la visibilité s_{ij} — laquelle correspond pour le PVC à l'opposé de la distance entre les villes.

$$P_{ij} = \begin{cases} \frac{{\tau_{ij}}^{\alpha} {s_{ij}}^{\beta}}{\sum_{(i,l)\in\Omega} {\tau_{il}}^{\alpha} {s_{il}}^{\beta}} & \text{si } (i,j) \in \Omega \\ 0 & \text{sinon} \end{cases} \tag{3.1}$$

avec Ω la liste des villes qu'il reste à parcourir.

La mise à jour des phéromones s'effectue lorsque toutes les fourmis artificielles ont parcouru toutes les villes (contrairement aux fourmis réelles qui font la mise à jour au fur et à mesure de leurs déplacements). Cette mise à jour dépend du coefficient d'évaporation ρ, qui

permet de représenter l'évaporation naturelle des phéromones, et du coefficient de renforcement $\Delta\tau_{ij}$ donné par la formule 3.2. Ainsi toutes les fourmis participent bien au renforcement des phéromones, mais avec plus ou moins d'insistance selon la qualité de la solution qu'elles ont générée — là encore il s'agit d'une possibilité offerte aux fourmis artificielles par rapport à leurs homologues réelles.

$$\Delta\tau_{ij}^{f} = \begin{cases} \frac{1}{L^f} & \text{si } (i,j) \in S^f \\ 0 & \text{sinon} \end{cases} \tag{3.2}$$

Des variantes ont ensuite permis d'améliorer l'algorithme AS, comme le Max-Min Ant System [126] pour borner la quantité de phéromones sur le chemin, ou encore le AS_{rank} [16] dans lequel le renforcement des phéromones tient compte du coût de la solution générée par la fourmi mais aussi de son rang par rapport aux coûts des solutions des autres fourmis. Par ailleurs, l'algorithme Ant Colony System [49] (ACS) présente une autre variante de résolution du PVC par des fourmis artificielles. Dans cet algorithme, seule la meilleure fourmi de chaque itération renforce les chemins qu'elle a utilisés pour construire sa solution. La mise à jour des phéromones ne s'effectue plus qu'à partir du meilleur chemin trouvé. De plus, une heuristique locale peut être ajoutée pour améliorer les solutions des fourmis.

3.3.3 Heuristique ACO

Désormais, l'heuristique ACO (*Ant Colony Optimization*) regroupe toutes les variantes issues de cette rencontre fourmis-phéromones-optimisation et aborde de nombreux problèmes combinatoires plus ou moins proches du PVC [52]. Cette heuristique peut de manière générale être présentée comme sur l'algorithme 2.

Algorithme 2 Ant Colony Optimization (ACO)

```
Initialiser les valeurs des phéromones τ_ij
Répéter
    Construction des solutions par les fourmis
    Mise à jour des phéromones
    Exécution d'une action centralisée
Jusqu'à la validation du test d'arrêt
Retourner la meilleure solution trouvée depuis le début
```

Tout d'abord, la fourmi se déplace sur les nœuds du graphe représentatif du problème pour construire sa solution. Elle est pour cela influencée par des décisions locales probabilistes qui tiennent compte à la fois de la connectivité du graphe et de l'intensité des traces de phéromones déposées par ses congénères dans les précédentes itérations. Ensuite elle doit annoter les arcs du graphe pour aider les autres fourmis à trouver le meilleur résultat en y déposant des phéromones. Enfin, il peut être utile d'exécuter certaines actions centralisées qui ne peuvent être effectuées par les fourmis elles-mêmes de manière individuelle, comme la collecte de la meilleure solution ou encore son coût.

3.4 Résolution de problèmes d'optimisation combinatoire par les fourmis artificielles

Nous allons ici nous intéresser à la résolution de problèmes qui peuvent se modéliser par un graphe. Tout d'abord nous aborderons un problème classique d'optimisation combinatoire se rapprochant du PVC, le « Vehicle Routing Problem » (VRP). Puis nous présenterons deux autres problèmes, à savoir le « Minimum Spanning Tree » (MST) et le « Keyboard Arrangement Problem » (KAP), que nous appliquerons au domaine du handicap dans les chapitres suivants.

Rappelons tout d'abord quelques notations et notions élémentaires. Soit $G = (S, A)$ un graphe, avec S un ensemble de sommets et A un ensemble de couples de sommets. Si les couples de sommets de A sont ordonnés alors ils sont appelés « arcs » et le graphe est dit « orienté » ; tandis que s'ils sont non-ordonnés, ils sont nommés « arêtes » et le graphe est « non-orienté ». Chaque graphe peut être représenté par une matrice d'adjacence qui sera symétrique dans le cas de graphe non-orienté. Les graphes permettent de modéliser divers problèmes du monde réel : réseau routier G=(villes, routes), site web G=(pages, liens), réseau internet G=(serveurs et utilisateurs, interconnexions), assignation des touches sur un clavier G=(touches, emplacements).

3.4.1 Le « Vehicle Routing Problem » (VRP)

Le VRP [48] est un problème consistant à trouver la ou les tournées permettant de relier un dépôt à un ensemble de clients, et ce en minimisant les coûts liés aux déplacements. Il peut se modéliser à l'aide d'un graphe de $n+1$ nœuds, où le nœud 0 correspond au dépôt et les nœuds 1 à n correspondent à des clients. Ce problème ressemble au PVC avec plusieurs voyageurs de commerce se partageant les villes à visiter, ou, à l'inverse, le PVC est un VRP avec un seul camion (de capacité illimité).

De nombreuses variantes à ce problème existent comme le « Multiple Depot VRP » (MDVRP) où l'entreprise possède plusieurs dépôts, le « VRP with Pick-up and Delivery » (VRPPD) où certains clients doivent également renvoyer des objets au dépôt, le « Split Delivery VRP » (SDVRP) où un client peut être livré par plusieurs véhicules, ou encore le « Capacitated VRP » (CVRP) où chaque véhicule ne peut transporter plus d'une certaine quantité de marchandise.

Dans le CVRP, tous les véhicules sont identiques et ils ont chacun une capacité de chargement noté W. De plus, chaque client a une demande connues q_i (avec $i = 1, .., n$) qui doit être satisfaite en une seule fois — les livraisons partielles ne sont effectivement pas autorisées. De ce fait, chaque client doit être livré une seule et unique fois. Remarquons ici que le PVC est un cas particulier du CVRP. En effet, si nous considérons que la somme des demandes de tous les clients est inférieure à la capacité d'un camion — ou que la capacité W est infinie —, la tournée peut être effectuée par un seul camion qui doit partir du dépôt, se rendre chez tous les clients par le chemin le plus court, pour finalement retourner au dépôt.

Des méthodes exactes permettent de résoudre le problème du VRP ou du CVRP lorsqu'il comporte peu de clients ([31] jusqu'à 25 clients ou [76] jusqu'à 60 clients). Avec des méta-heuristiques, des résultats approchés pour une plus grande quantité de clients [14][25][23] sont obtenus. Les fourmis artificielles participent elles aussi à la résolution de ce problème. Le PVC et le VRP étant très liés, les méthodes de résolution du VRP [16][17][113][111] sont fortement inspirées de celles du PVC, à savoir le Ant System.

Dans l'algorithme AS_{rank} [16], la mise à jour des phéromones s'effectue pour toutes les fourmis et dépend du rang de la solution générée par la fourmi par rapport à celui de ses congénères. L'algorithme [17] réutilise la formule de mise à jour des phéromones donnée dans l'AS_{rank}. De plus, la visibilité s_{ij} de la formule de probabilité ne correspond plus à la distance entre les villes comme dans le Ant System, mais à l'économie réalisée en reliant deux villes dans un même circuit plutôt qu'en les séparant dans deux circuits différents.

Remarquons que les fourmis artificielles sont également utilisées pour résoudre des variantes du VRP [62][110].

3.4.2 Le « Minimum Spanning Tree » (MST)

Le PVC est un problème complexe alors que sa restriction consistant à trouver l'arbre couvrant de poids minimum (ACM, ou « Minimum Spanning Tree » MST) à partir d'un graphe est un problème facile qui peut être résolu par l'algorithme de Prim [106] en $O(n^2)$ (avec n le nombre de sommets du graphe). Précisons que l'ACM est une restriction du PVC car la majorité des arêtes d'un arbre couvrant de poids minimum appartient à la tournée optimale du PVC. Ainsi la longueur de l'ACM constitue une borne inférieure au coût de la solution du PVC.

Par définition, un arbre couvrant de poids minimal (ACM) est un graphe connexe qui n'a pas de cycle, qui possède $n - 1$ arêtes si n est le nombre de sommets, et dont la somme des poids de l'ensemble de ses arêtes est minimale. Dans l'algorithme de Prim, le premier sommet du graphe marqué est fixé aléatoirement ; dès lors, à chaque itération il connecte un sommet non marqué de G à un sommet marqué (c'est-à-dire déjà placé dans l'arbre en construction). L'arbre va ainsi grossir jusqu'à ce qu'il couvre tous les sommets du graphe. A chaque étape, le nouveau sommet y à marquer est tel que y est adjacent à un sommet x marqué et que l'arête (x, y) est celle de plus faible poids. Sur le graphe donné en exemple sur la figure 3.2 (a), si le sommet a est le premier choisi, l'algorithme de Prim permet au sommet b, qui est celui le plus proche de a, de rejoindre l'ACM en construction, puis c'est au tour du sommet c d'être rattaché au sommet b, etc... A la fin de l'exécution, l'ACM illustré par les figures 3.2 (b) et (c) est obtenu.

Le « Generalized Minimum Spanning Tree » (GMST)

Dans le problème « Generalized Minimum Spanning Tree » (GMST), les sommets du graphe G sont regroupés en cluster, chacun contenant au minimum un sommet du graphe.

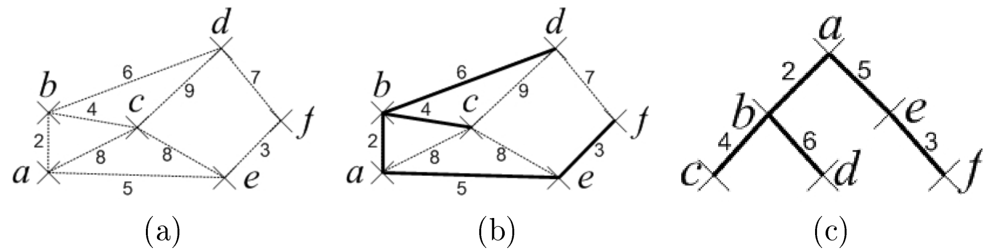


FIG. 3.2 – Exemple d'ACM représenté sous deux formes différentes (b) et (c) à partir d'un graphe (a).

Le but du GMST est de construire un arbre de poids minimum contenant au moins un sommet de chaque cluster.

Pour résoudre ce problème difficile, des métaheuristiques telles que des algorithmes génétiques ont été proposées [54]. L'heuristique ACO peut également être utilisée pour résoudre ce problème. Lors de la construction de l'arbre la fourmi doit choisir sa prochaine destination en utilisant la probabilité donnée par la formule 3.1 (page 90). Elle utilise pour cela les phéromones laissées par ses congénères ainsi que la visibilité s_{ij} représentant dans le cas du GMST l'inverse du coût du lien entre le nœud i et le nœud j, sachant que ce dernier n'a pas encore été visité. Ainsi les fourmis construisent un arbre jusqu'à ce que tous les clusters aient au moins été visités une fois. Afin d'améliorer les performances de cet algorithme pour résoudre le problème du GMST, Shyu S.J. et al. [123] propose l'algorithme Ant-Tree où la visibilité s_{ij} dépend du cluster d'appartenance des nœuds i et j. Ainsi si j appartient à un cluster qui n'a pas encore été visité, la visibilité s_{ij} est la même que lors de l'application de l'algorithme ACO au GMST. Par contre, si j appartient au même cluster que i ou s'il appartient à un cluster déjà visité, la visibilité s_{ij} est réduite. Ainsi, les fourmis sont encouragées à visiter dans un premier temps tous les clusters.

Le « Degree-Constrained Minimum Spanning Tree » (d-MST)

Le but du « Degree-Constrained Minimum Spanning Tree » (d-MST) est de construire un arbre de poids minimum tel que chaque nœud possède un degré maximum, c'est-à-dire que chaque nœud à un nombre de nœuds fils maximum fixé.

Narula S.C. et Ho C.A. [97] propose une méthode pour résoudre ce problème s'appuyant sur l'algorithme de Prim et appelée d-Prim. A chaque étape de la construction de l'arbre, ils vérifient que le degré maximum n'est pas atteint. S'il est atteint alors le nœud de l'arbre en construction n'apparait plus dans la liste des nœuds marqués auxquels on peut connecter d'autres nœuds. De même Caccetta L. [19] a proposé une méthode s'appuyant sur l'algorithme de Kruskal [73], ce dernier permettant, tout comme celui de Prim, de résoudre le MST. Pour résoudre ce problème des algorithmes utilisant les fourmis artificielles ont également été proposés. Dans l'algorithme de Bau et al. [8], chaque fourmi construit un arbre en utilisant pour ce faire la probabilité donnée par l'équation 3.1 (page 90) qui lui permet de

trouver le meilleur nœud non marqué à ajouter à l'arbre en construction. Elle est pour cela influencé par la quantité de phéromones entre les nœuds et par la visibilité s_{ij} qui correspond à l'inverse du coût du lien entre les nœuds i et j. Comme dans l'algorithme d-Prim, la contrainte concernant le degré maximum de chaque nœud est vérifiée à chaque ajout d'un nouveau nœud.

Le « Capacitated Minimum Spanning Tree » (CMST)

Le problème « Capacitated Minimum Spanning Tree » (CMST) consiste à connecter un ensemble de nœuds clients à un nœud racine en minimisant le coût de la construction tout en tenant compte de la capacité des liens établis entre les clients (par exemple lors de la délivrance d'un signal aux clients dans les réseaux de télécommunication).

Ce problème a été résolu par des méthodes exactes [5] pour de petites instances mais aussi par des métaheuristiques [122][104] pour un nombre de clients plus important. A ce propos, Reimann et Laumann [112] ont proposé un algorithme utilisant des fourmis artificielles pour résoudre le CMST. Ils se basent sur deux observations qui sont les suivantes. D'une part, le problème CMST peut être considéré comme une extension du CVRP. Dans les deux cas les nœuds ont besoin d'être regroupés pour satisfaire à la contrainte de capacité. De ce fait ils proposent d'utiliser les fourmis artificielles pour résoudre le CVRP (c'est-à-dire en réalisant des circuits de nœuds, voir section 3.4.1). Pour ce faire, les fourmis utilisent la formule de probabilité de l'équation 3.1 (page 90), où s_{ij} représente l'économie réalisée lorsque i et j sont placés dans un même circuit, plutôt que dans deux différents. Cependant, une solution de chaque regroupement correspond dans le CVRP à un circuit (de villes), alors que dans le CMST il s'agit d'un sous-arbre (de clients). La solution du CVRP constitue alors une borne supérieure au CMST. La solution générée par les fourmis constitue ainsi une première étape pour résoudre le CMST. D'autre part, Reimann et Laumann observent qu'une fois les regroupements de nœuds effectués, le sous-problème consiste à réaliser un ACM. Les sous-problèmes peuvent donc être résolus en un temps polynomial avec l'algorithme de Prim. Ainsi l'algorithme de Reimann et Laumann commence par construire une solution au problème CVRP à l'aide de fourmis artificielles, pour ensuite appliquer l'algorithme de Prim aux sous-circuits définis.

3.4.3 Le « Keyboard Arrangement Problem » (KAP)

Le « Keyboard Arrangement Problem » (KAP), ou problème d'organisation des touches sur un clavier a été présenté dans la section 2.3. Nous y avons vu que Eggers and al. [56] ont généré un clavier avec un placement des touches optimisé pour une langue donnée qui respecte certains critères d'ergonomie pour la saisie à dix doigts sur un clavier ergonomique. Pour trouver l'organisation des touches ils ont utilisé un algorithme de fourmis artificielles. Nous allons ici présenter le rôle des fourmis artificielles dans la génération de leur clavier.

Au départ chaque fourmi de la colonie se voit assigner un clavier vide et un texte. La fourmi doit lire le texte et trouver le meilleur placement sur le clavier du caractère sur

lequel elle se trouve, jusqu'à ce qu'elle ait placé tous les caractères du texte. Le choix de ce placement est influencé par la probabilité donnée par l'équation 3.1 (page 90), où dans le cas du KAP :

- Ω est l'ensemble des emplacements sur le clavier qui ne sont pas encore assignés ;
- τ_{ij} est la quantité de phéromones entre la lettre i et l'emplacement j ;
- s_{ij} représente la visibilité, soit l'évaluation de la qualité ergonomique du clavier si la lettre i est placée à l'emplacement j ;
- α et β sont deux coefficients pour favoriser l'expérience des fourmis par rapport à la visibilité ou inversement.

Ainsi les fourmis auront plus de chance de placer la lettre i sur l'emplacement k si τ_{ij} et s_{ij} sont importants.

Lorsque toutes les fourmis ont généré une organisation des touches, les phéromones entre les lettres et les emplacements sont mises à jour en utilisant la formule de l'algorithme AS, ou plus précisément sa variante donnée dans l'algorithme AS_{rank} [16] consistant à déposer une quantité de phéromones dépendant de la qualité de la solution générée par la fourmi par rapport à celles de ces congénères.

Rappelons que ce travail a permis de montrer que les fourmis artificielles étaient capable d'optimiser l'arrangement des touches sur un clavier ergonomique en donnant de meilleurs résultats que l'organisation Dvorak pour la saisie à dix doigts.

3.5 Conclusion

Les algorithmes de colonies de fourmis sont considérés comme des métaheuristiques dans lesquelles chaque fourmi construit une solution en tenant compte des marquages laissés par les précédentes fourmis pour optimiser leur recherche. Pour les versions adaptées aux problèmes combinatoires, la construction de la solution s'effectue de manière itérative, à l'aide d'une probabilité utilisant une mémoire locale de l'environnement.

Nous avons présenté dans ce chapitre l'heuristique ACO dans laquelle les fourmis utilisent les phéromones comme moyen de communication au sein de la colonie, ainsi que son utilisation pour la résolution de certains problèmes d'optimisation combinatoire que nous appliquerons par la suite au domaine du handicap. Précisons que les phéromones ne sont pas le seul moyen utilisé par les fourmis pour communiquer et que le fourragement n'est pas le seul comportement intéressant de ces insectes sociaux. La résolution de problèmes d'optimisation n'est donc pas la seule application de la source d'inspiration que représentent les fourmis, la classification [93] ou la robotique mobile [9] en sont des exemples.

Deuxième partie

Nos contributions à l'accessibilité du web pour les personnes handicapées

Chapitre 4

Outils d'aide à l'accessibilité

Ce chapitre présente divers outils d'aide à l'accessibilité. D'une part, nous présentons un outil d'aide à la création de site web accessible réalisé en collaboration avec le Conseil Général d'Indre-et-Loire pour les webmestres de ses collectivités locales. D'autre part, au vue des difficultés rencontrées par les personnes handicapées pour lire rapidement ou pour saisir des informations, même sur un site accessible, nous présentons tout d'abord un générateur de braille abrégé, puis une nouvelle approche utilisant des fourmis artificielles pour générer des claviers virtuels minimisant les déplacements pour saisir un type de texte donné.

4.1 Introduction

Malgré la loi votée en 2005 en France préconisant l'accessibilité des sites web publics, encore trop peu de sites sont accessibles. Il peut y avoir plusieurs raisons pour expliquer cela. Tout d'abord, certains propriétaires de sites web ne se sentent pas concernés par les difficultés rencontrées par les personnes handicapées, et ne voient alors pas la nécessité de rendre leur site accessible. Ensuite, certains webmestres dénoncent le manque d'informations techniques sur la manière de rendre un site accessible. Finalement, d'autres webmestres dit « débutants » se sentent quant à eux submergés par les diverses normes et les nombreux outils tels que les validateurs, et ne savent pas comment débuter. Dans tous les cas, cela montre un manque d'information, mais surtout des informations adaptées aux connaissances des webmestres.

En collaboration avec le Conseil Général d'Indre-et-Loire, nous avons réalisé un outil d'aide à la création et à la migration de sites web accessibles. Cet outil est destiné aux petites collectivités locales qui possèdent bien souvent un site web mis en place par un membre bénévole, que nous qualifierons de webmestre « débutant ». Cet outil, qui à la particularité de proposer une démarche personnalisée, sera présenté dans la section 4.2 de ce chapitre.

Toutefois les webmestres ne sont pas les seules personnes à qui sont destinés les outils d'aide présentés dans ce chapitre. Les sections 4.3 et 4.4 traiteront de deux dispositifs d'aide — ici des aides techniques — destinés à accélérer l'accès en lecture ou en écriture pour les personnes handicapées. D'une part, nous avons établi un générateur de braille abrégé pour accélérer la lecture des pages web pour les handicapés visuels. D'autre part, nous avons réalisé un clavier virtuel, destiné à accélérer la vitesse de saisie et diminuer la fatigue des membres antérieurs ressentie par les personnes handicapées, qui propose de réorganiser les touches du clavier selon le type de texte à saisir.

4.2 Site web d'aide à la migration

Dans le but d'aider les collectivités locales dans le domaine de l'accessibilité, et en collaboration avec le Conseil Général d'Indre et Loire, nous avons élaboré un « outil d'aide à la création et à la migration de sites Internet accessibles » (figure 4.1). Cet outil a été réalisé suite à une réflexion sur la démarche méthodologique de migration et d'accompagnement vers l'accessibilité du web.

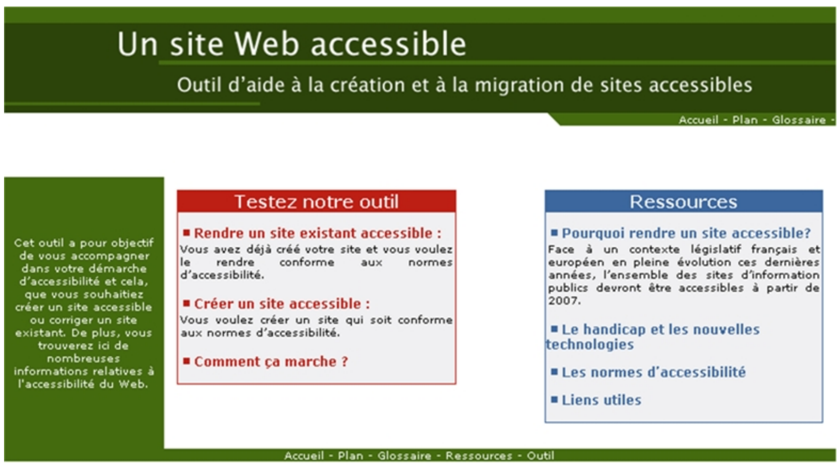


FIG. 4.1 – Page d'accueil de notre outil d'aide à la migration et à la création.

Il permet d'informer et d'aider quiconque souhaite rendre ou créer un site accessible. Toutefois, comme nous l'avons vu dans la section 1.4.2 (page 40), des guides existent déjà sur Internet, et résument ce qu'est l'accessibilité. Le tableau 4.1 reprend le tableau 1.4 de la page 42 — comparant les différents guides existant — et y intègre notre outil.

En supplément des fonctionnalités proposées par les autres guides, notre outil initie à l'utilisation des validateurs automatiques et explique les erreurs les plus fréquentes. Il possède de plus une mission plus pratique qui consiste à offrir un véritable mode d'emploi pour rendre ou créer un site accessible.

4.2.1 Les ressources nécessaires pour bien démarrer

Afin de bien débuter dans la démarche d'accessibilité, notre outil propose une partie « Ressources » (figure 4.2) qui aborde :

- le contexte législatif
  Il informe sur la notion d'accessibilité, en expliquant pourquoi il convient de rendre un site accessible, tout en présentant le contexte législatif en France et en Europe.
- le handicap visuel et les aides techniques
  Mise à part l'obligation de rendre les sites web accessibles imposée par la loi de février 2005, il nous incombe de rappeler l'utilité première d'un site accessible. Pour cela,

	La grange	Voir+	BrailleNet	Accès pour tous	Notre outil
Informations sur le handicap et les aides techniques	x		x		x
Contexte législatif		non actualisé	non actualisé		x
Lien vers les recommandations officielles (HTML, CSS, WCAG)	x	x	x	x	x
Explication des recommandations officielles			x		x
Lien vers des outils de validation automatique	x	x	x	x	x
Fiches techniques	x		x	x	x
Validateur intégré				x	x
Démarche d'accessibilité				générale	personalisée

TAB. 4.1 – Comparaison de différents guides d'accessibilité avec notre outil.

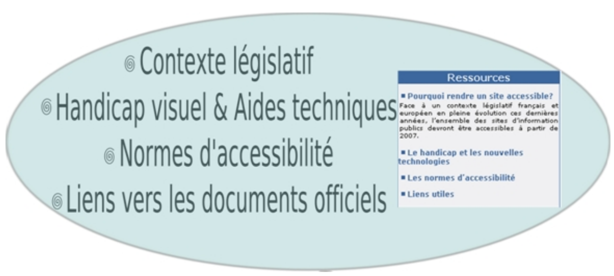


FIG. 4.2 – Les ressources.

les différents types de déficiences visuelles sont présentés, ainsi que les difficultés que cela engendre lors de la navigation sur le web. De plus, les aides techniques utilisées couramment par les non-voyants et les malvoyants sont décrites.

- les normes d'accessibilité
  Les recommandations officielles sont ici détaillées et, pour chacune, des outils de validation automatique sont proposés.
- les liens vers les recommandations officielles
  Dans cette partie sont listées les principales références. On y trouve par exemple des liens vers les recommandations du W3C ainsi que vers des outils de validation automatique.

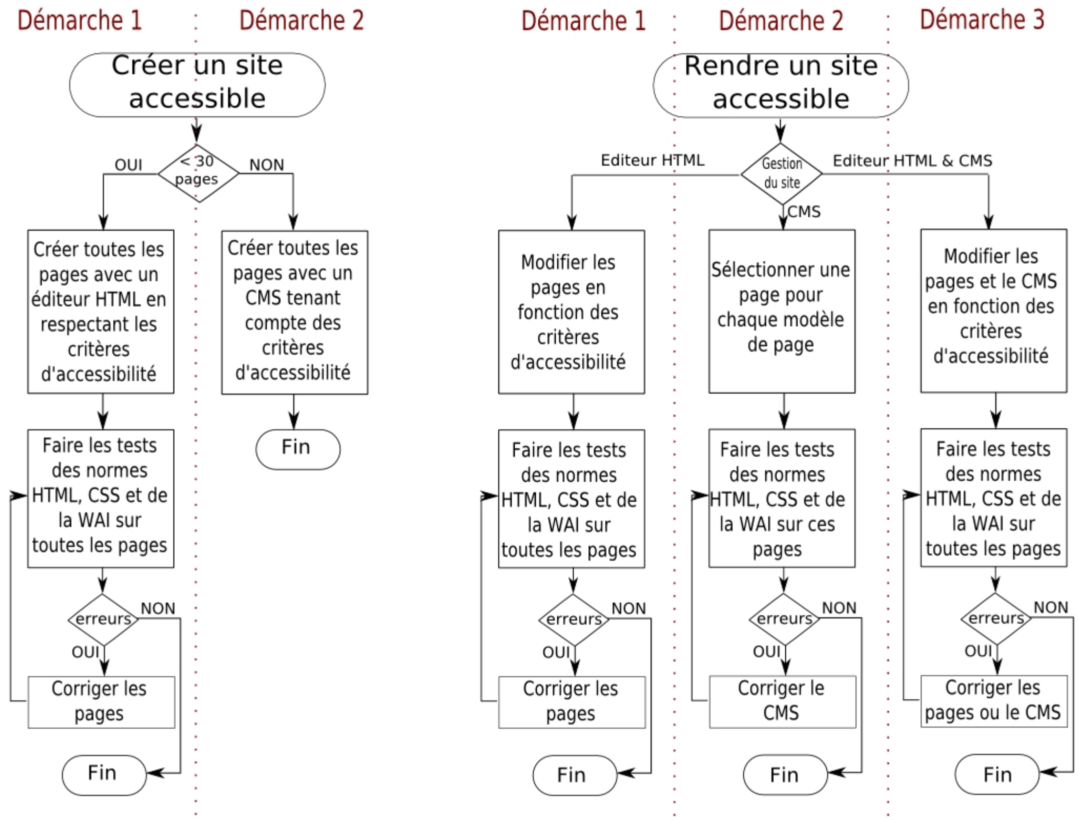
4.2.2 Mode d'emploi pour la création ou la migration de site accessible

Notre outil offre de plus une partie plus pratique (figure 4.3) concernant la réalisation d'un site accessible ou la migration d'un site existant. Rappelons qu'il est destiné aux webmestres des collectivités locales, dont les connaissances — et de ce fait les technologies utilisées — sont très variables. Pour les guider dans leur démarche, nous proposons un mode d'emploi détaillé contenant les différentes étapes à suivre, selon le type de site web — celui-ci étant identifié grâce à une série de questions simples posées au webmestre.



Fig. 4.3 – Accéder au mode d'emploi personnalisé.

Dans un premier temps, nous proposons deux démarches (illustrées par la figure 4.4 (a)) pour créer un site web accessible. Elles tiennent compte de la taille du site, critère principal de différenciation des sites web des collectivités locales. Ainsi, la première démarche (figure 4.4 (a), démarche 1) permet de créer un site contenant peu de pages. Dans ce cas il faut les créer une à une puis vérifier que chacune respecte les normes. Afin de ne pas avoir trop d'erreurs à corriger, il convient d'être informé sur les recommandations avant de commencer à rédiger les pages. A ce stade, le webmestre peut utiliser divers petits outils pratiques, que nous présenterons dans la section 4.2.3, comme par exemple des fiches techniques. Une fois toutes les pages créées en respectant au mieux les recommandations, l'outil propose de tester le code HTML, les feuilles de style, et pour finir l'accessibilité. Des validateurs sont alors proposés ainsi qu'une aide à la correction des principales erreurs (détaillée dans la section 4.2.3). La deuxième démarche (figure 4.4 (a), démarche 2) concerne la création d'un site conséquent ou bien nécessitant de fréquentes mises à jour. Dans ce cas, notre outil est le seul à conseiller l'utilisation d'un système de gestion de contenu (Content Management System (CMS)) pour créer le site. Cela permet de décentraliser les mises à jour de contenu et d'alléger la charge de travail du webmestre. En outre, le choix d'un CMS peut être difficile et il faut en sélectionner un qui offre le maximum de fonctionnalités attendues pour le site — et bien évidemment un qui puisse produire du code conforme aux normes d'accessibilité. Ainsi, pour aider dans le choix du CMS, notre outil propose des liens vers plusieurs CMS « accessibles » (SPIP AGORA, TYPO3...).



(a) (b)

FIG. 4.4 – Mode d'emploi pour créer (a) ou rendre (b) un site web accessible.

Dans un deuxième temps, trois démarches sont proposées pour rendre un site web accessible (figure 4.4 (b)). Dans ce cas, elles tiennent compte du mode de gestion du site web. Tout d'abord, si le site est statique (figure 4.4 (b), démarche 1), toutes les pages HTML doivent être testées séparément. La figure 4.5 illustre cette démarche telle qu'elle est présentée dans notre outil. Ensuite, si le site est géré par un CMS (figure 4.4 (b), démarche 2), le webmestre doit le paramétrer pour qu'il produise du code accessible puis des tests sur certains modèles de pages permettent de vérifier l'accessibilité du site. Finalement, s'il est géré par un CMS et en même temps par un éditeur HTML (figure 4.4 (b), démarche 3), un mélange des deux premières démarches permet de le rendre accessible.

4.2.3 Divers outils pratiques

Notre logiciel fournit également un ensemble d'outils pratiques (figure 4.6) qui peuvent être utilisés seuls ou bien en s'appuyant sur les différentes démarches pour créer ou rendre un site accessible.

Pour commencer, notre outil propose un ensemble de 11 fiches techniques regroupant les recommandations par thème. Chacune est composée d'explications et d'exemples d'appli-

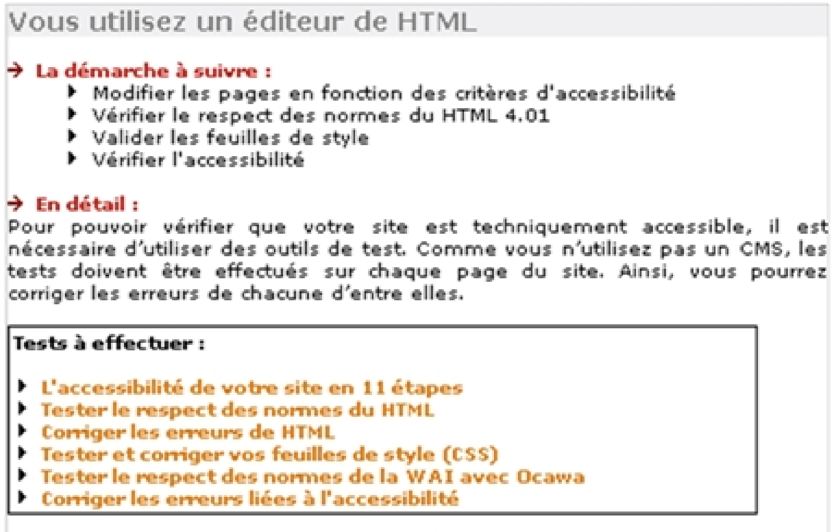


FIG. 4.5 – Démarche pour rendre un site (géré avec un éditeur HTML) accessible.

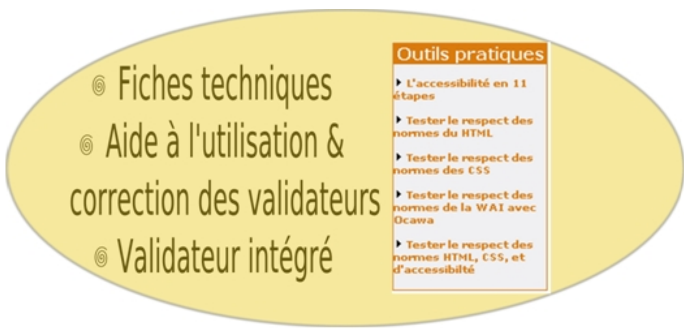


FIG. 4.6 – Les divers outils.

cation. Il est possible de lire simplement ces fiches techniques ou alors de créer un compte mémorisant les points qui ont déjà été vérifiés et ceux qu'il reste à contrôler. La figure 4.7 présente la fiche technique concernant les liens ; le cadre à droite affiche l'avancement dans les 11 fiches techniques (dans le cas d'un utilisateur qui a créé un compte). Dans cette illustration, on peut voir que la fiche technique concernant les éléments obligatoires (doctype, langue ...) et celle concernant les tableaux ont été validées. Cela signifie que l'utilisateur a indiqué qu'il a vérifié et/ou corrigé ses pages HTML conformément aux indications contenues dans ces deux fiches.

Ensuite, notre outil dispose d'une aide à l'utilisation des validateurs automatiques et à la correction des erreurs décelées par ces derniers. Pour rendre un site web accessible, il faut d'abord valider le code HTML et CSS, puis vérifier les normes d'accessibilité. Il est possible de tester celles-ci grâce à de nombreux validateurs automatiques. Notre outil propose à l'utilisateur un descriptif ainsi qu'une aide à l'utilisation d'un validateur pour chacun des tests à réaliser. De ce fait l'utilisateur n'a pas de surprise en arrivant sur la page

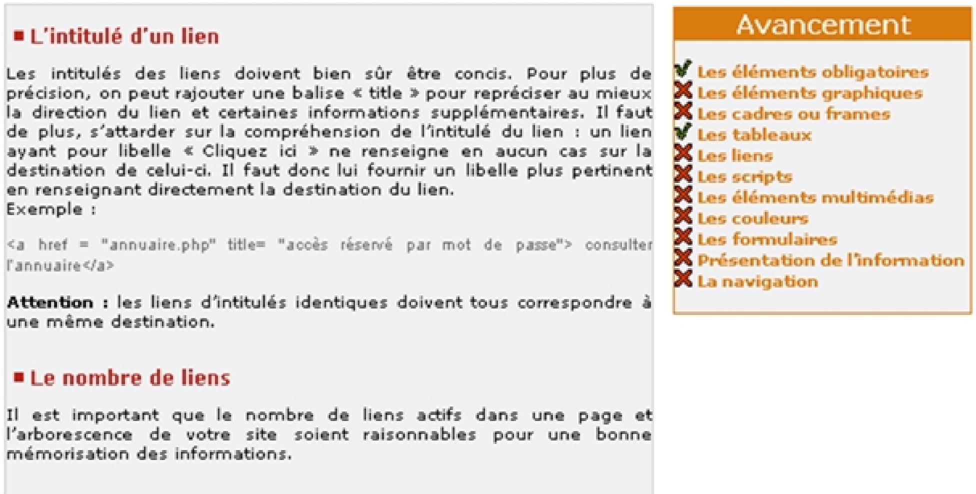


FIG. 4.7 – Avancement dans les 11 fiches techniques.

du validateur puisqu'il sait ce qu'il lui reste à faire pour lancer la vérification. Dès que le validateur a traité une des pages du site, l'utilisateur se trouve face à une liste d'erreurs. Il doit alors commencer à corriger les erreurs décelées par le validateur. Afin de l'aider lors de la correction, notre outil liste les erreurs les plus fréquentes, et, si l'utilisateur possède dans son site une de ces erreurs, il peut alors visualiser une aide à la correction et un exemple de code. La figure 4.8 illustre l'aide à la résolution d'une erreur HTML concernant l'attribut `alt`.



FIG. 4.8 – Aide à la résolution de l'erreur HTML : « Required attribute ALT not specified ».

Enfin, un validateur des normes d'accessibilité est intégré à notre outil. Il utilise le validateur OCAWA et offre par conséquent une détection d'erreurs de qualité. Ainsi l'utilisateur de notre outil peut tester les pages de son site sans avoir besoin de se rendre sur le site d'OCAWA puisque les pages sont soumises automatiquement à OCAWA, et ce de manière transparente pour l'utilisateur. Concrètement, notre outil soumet à OCAWA l'adresse de

la page à auditer, récupère le rapport d'erreurs et enregistre toutes les erreurs ainsi que leurs types dans sa base de données. Afin de rendre notre outil plus rapide, OCAWA nous a autorisés à accéder à leur serveur pour récupérer directement le fichier XML contenant les erreurs. Cette collaboration avec Urbilog, partenaire de France Telecom R&D pour le développement de l'outil OCAWA, nous a également permis de ne pas être dépendant des changements de version de l'interface d'OCAWA. Nous avons intégré le validateur à notre outil pour permettre à l'utilisateur, d'une part de sauvegarder l'état d'avancement de la validation — et ainsi pouvoir reprendre le processus de mise en conformité ultérieurement sans avoir à tester de nouveau la page —, et d'autre part de visualiser la liste des erreurs par type et non selon l'ordre d'apparition dans la page web testée. Le regroupement des erreurs par thème évite en effet à l'utilisateur de perdre son enthousiasme face à la multitude d'erreurs à corriger (car bien souvent elles sont répétitives). Il a ainsi une vision générale des erreurs qu'il lui reste à corriger. La figure 4.9 montre le résultat de l'analyse d'une page. On observe deux types d'erreurs. Le premier se répète deux fois : sur la figure, l'utilisateur a ouvert cette erreur donc les lignes où elle apparaît sont visibles. Le deuxième type d'erreur se répète 42 fois. Avec un validateur classique, l'utilisateur aurait directement vu une longue liste d'erreurs alors qu'ici il sait tout de suite qu'il n'a que deux types d'erreurs à savoir corriger.

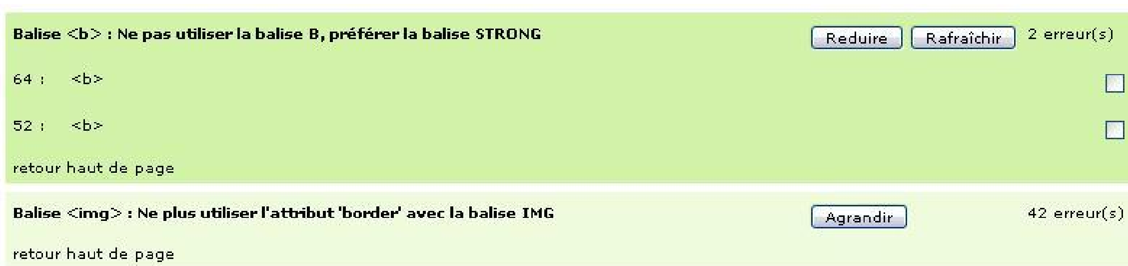


FIG. 4.9 – Résultat de l'analyse d'une page avec le validateur intégré à notre outil.

4.2.4 Application

Le conseil général d'Indre-et-Loire gère les affaires du département et se doit d'assurer un certain nombre de missions qui touchent à la vie quotidienne des citoyens. Un de ses domaines de compétence concerne l'action sociale et la santé, notamment l'insertion des personnes handicapées. Avec l'émergence en France de la loi concernant l'accessibilité numérique, le conseil général d'Indre-et-Loire a tenu, préalablement à la mise aux normes de son site internet, à participer activement à la diffusion de ce qu'est l'accessibilité et à la manière d'y parvenir. Par conséquent, c'est en collaboration avec le conseil général d'Indre-et-Loire que notre outil d'aide à la création et à la migration de site internet accessible est né. Grâce à cet outil, le conseil général d'Indre-et-Loire est en mesure d'aider l'ensemble de ses collectivités locales dans leur démarche de migration vers l'accessibilité. Cet outil s'adresse à quiconque souhaite rendre un site accessible et cela en s'adaptant au niveau de l'utilisateur en termes

de création de site. Le but de cet outil étant bien entendu de participer activement à la large diffusion de ce qu'est l'accessibilité et de proposer les moyens d'y parvenir.

Une application concrète [37] de notre outil « d'aide à la création et à la migration de sites web accessibles » a été réalisée sur le site web de la mairie de Chambray-lès-Tours (37). Par exemple, la page d'accueil de ce site comportait 44 erreurs. Ces erreurs se répétant, elles ont été regroupées en seulement 2 types d'erreurs. Pour celles-ci notre outil a proposé des explications et des corrections. Ainsi les 44 erreurs de la page d'accueil ont pu être facilement corrigées, puisqu'il ne s'agissait en fait que de 2 erreurs répétitives.

4.3 Traducteur en braille abrégé

La lecture des informations peut être plus ou moins rapide selon les capacités des personnes handicapées visuelles et selon les aides techniques qu'elles utilisent. Il est en effet possible de régler la vitesse du débit de parole d'une synthèse vocale, ou encore de faire défiler les lignes d'un clavier braille à un rythme défini par l'utilisateur. Cependant si le débit reste trop lent lors de l'utilisation d'un clavier braille, on peut envisager de lui faire transcrire non pas du braille intégral mais du braille abrégé. La méthode toujours en vigueur actuellement est celle de 1955 qui exige environ 4 années d'apprentissage et la connaissance de nombreuses règles. Ce système a été révisé en 2000 dans la thèse de Carmen Fontaine [59] et en 2004 par Evelyne Kommer [72].

Afin de nous familiariser avec les aides techniques comme le clavier braille, nous avons créé une application permettant la transcription d'un texte français en braille abrégé, la lecture se faisant sur un terminal braille adapté. La méthode consiste à découper un texte en mots, à les soumettre à un traitement de locutions et ensuite à vérifier les règles de compression. Ce procédé permet de réduire les ouvrages imprimés en braille d'une part et d'autre part d'augmenter la vitesse de lecture. Ainsi cet outil pourra être intégré à d'autres que nous avons créés pour accélérer la lecture sur Internet.

4.3.1 Méthode de transcription

Comme nous l'avons précisé dans la section 2 (page 19), un texte en braille abrégé est composé de mots — contenant une ou plusieurs contractions — de symboles, et de mots en intégral.

Pour transcrire un texte en braille abrégé, il faut tout d'abord détecter les locutions. Une locution correspond à une séquence de mots ayant une abréviation définie. Les locutions ainsi que leurs abréviations sont répertoriées dans un dictionnaire. Ensuite, pour les mots ne correspondant pas à des locutions, il faut rechercher si une ou plusieurs abréviations sont possibles. Pour cela il faut dans un premier temps essayer de trouver une abréviation du mot complet, on parle alors de symbole, et si aucun symbole n'est trouvé, il faut dans un deuxième temps essayer de contracter le plus de lettres possible. Les algorithmes principaux de transcriptions en braille abrégé sont donnés dans l'annexe A. A l'instar des locutions, un dictionnaire des symboles est établi. Les règles de contraction permettent d'abréger des ensembles de lettres mais uniquement si cet ensemble appartient à une même syllabe.

Par définition, une syllabe est un son ou un groupe de sons prononcé par une seule émission de voix. Afin de détecter de manière automatique les différentes syllabes d'un mot, nous avons utilisé les règles de division suivantes :
- une consonne placée entre deux voyelles introduit une nouvelle syllabe ;
- quand deux consonnes sont placées entre deux voyelles, la première appartient à la syllabe précédente, la seconde à la syllabe suivante ;
- les groupes bl, cl, fl, gl, pl, br, cr, dr, fr, gr, pr, tr, vr qui contiennent un l ou un r, sont inséparables ;

– les groupes ch, ph, gn et th sont inséparables ;
– quand il y a trois consonnes consécutives à l'intérieur d'un mot, les deux premières terminent une syllabe, la dernière commence quant à elle une nouvelle syllabe ;
– les groupes inséparables cités précédemment commencent une syllabe.

Si on note V les voyelles, C les consonnes et « / » le caractère de séparation des syllabes, les règles citées précédemment peuvent être synthétisées comme suit :

1. $VCV \rightarrow V/CV$
2. $VCCV \rightarrow \begin{cases} VC/CV & \text{si CC n'est pas inséparable} \\ V/CCV & \text{sinon} \end{cases}$
3. $CCC \rightarrow \begin{cases} CC/C & \text{si la 2}^{\text{ème}} \text{ et la 3}^{\text{ème}} \text{ consonnes ne sont pas inséparables} \\ C/CC & \text{sinon} \end{cases}$

Prenons par exemple le mot « montagne » à découper en syllabes. A partir de la lettre « m » il n'est pas possible d'appliquer une règle. Il faut donc passer à la lettre suivante : le « o ». C'est une voyelle suivie de deux consonnes puis d'une voyelle. Il s'agit donc du cas VCCV (règle 2). Or comme « nt » ne fait pas partie des groupes inséparables, on doit découper le mot après le « n », ce qui donne mon/tagne. Par la suite, on observe encore une fois VCCV avec les lettres « agne » mais cette fois le groupe « gn » est inséparable donc on utilise le deuxième cas de la règle 2 qui permet de diviser après le « a ». On obtient donc le découpage mon/ta/gne.

4.3.2 Résultats expérimentaux

Le but de ce générateur de braille abrégé étant en partie d'être intégré à d'autres outils accélérant la vitesse de lecture sur Internet, nous avons évalué le temps d'exécution de notre algorithme de transcription, pour estimer si le temps d'exécution ne risquait pas de faire attendre l'internaute. Pour cela nous avons réalisé des tests sur plusieurs extraits d'œuvres dont les références sont les suivantes :

1. Molière (1622-1673) - *Le Bourgeois Gentilhomme* (1670) - Acte 2, Scène 4
2. Pierre Corneille (1606-1684) - *Le Cid* (1682) - Acte 1, Scène 1
3. Charles Perrault (1628-1703) - *Contes de ma mère l'Oye* (1697) - « Cendrillon »
4. Charles Perrault (1628-1703) - *Contes de ma mère l'Oye* (1697) - « Le Petit Chaperon Rouge »
5. Jean-Jacques Rousseau (1712-1778) - *Confessions*, Livre 12 (1789)
6. Victor Hugo (1802-1885) - *Notre-Dame de Paris* (1831)
7. Gustave Flaubert (1821-1880) - *Madame Bovary* (1857) - Chapitre 1
8. Alphonse Daudet (1840-1897) - *Le Petit Chose* (1868)
9. Jules Verne (1828-1905) - *Le Tour du Monde en 80 jours* (1872) - Chapitre 1

10. Marc Levy (1961-) - *Et si c'était vrai* (2001)

La figure 4.10 montre un exemple de résultat obtenu pour une partie du texte 1 de Molière. A savoir : à droite le texte original et à gauche ce même texte transcrit en abrégé.

Monsieur Jourdain	**¿mr Jourdain**
Apprenez-moi l'orthographe.	*¿ap!enz-mù l'Ñthoïaphe.*
Maître de philosophie	**¿maî)e d philosophie**
Très volontiers.	*¿)s v½.*
Monsieur Jourdain	**¿mr Jourdain**
Après, vous m'apprendrez l'almanach, pour savoir quand il y a la lune et quand il n'y en a point.	*¿a!, v m'ap!?ôz l'almanaà, ç savùr qd i y a ' lune ù qd i n'y? a pt.*
Maître de philosophie	**¿maî)e d philosophie**
Soit.	*¿sùt.*

FIG. 4.10 – Texte de Molière en intégral puis en abrégé.

Dans l'exemple de la figure 4.10, on peut voir qu'une lettre en majuscule est représentée en braille par deux caractères : le caractère « ¿ » précisant qu'il s'agit d'une majuscule et le caractère lui-même. On peut également trouver l'abréviation de symboles : « monsieur » possède pour symbole en braille abrégé « mr » ; « vous » devient « v » ; « pour » devient « ç »... A partir de cela, on comprend aisément que « Monsieur » se traduise par « ¿mr » en braille abrégé. De plus, on rencontre quelques contractions : dans le mot « apprenez », « pr » est suivi d'une voyelle et se contracte donc en « ! » ; « oi » se contracte quelque soit son emplacement en « ù » dans « moi », « savoir » et « soit ». Remarquons que « ù » est également le symbole du mot « et ».

Dans cet exemple, on observe aussi quelques caractères inhabituels comme « Ñ ». Ceux-ci ne font pas partie du braille abrégé mais ont été introduits pour les besoins de la programmation. En effet, l'alphabet braille offre 2^6 possibilités (soit 64 combinaisons), et le français n'en utilise que 50 (26 lettres, 10 signes de ponctuations et 14 signes comprenant 13 lettres accentuées et le « ç »). Il en découle 14 possibilités de symboles braille, appelés caractères « hors séries », qui sont utilisés essentiellement pour le braille abrégé et ne correspondent pas à des caractères utilisés dans la langue française. C'est pour cela que ces caractères hors séries ont été traduits par des caractères spéciaux dans notre application. Par exemple « or » se contracte en « Ñ », ce qui correspond en fait à un caractère braille non traductible en français. Le caractère « ¿ » représentant une majuscule fait d'ailleurs lui aussi partie des caractères hors séries.

Sur le tableau 4.2, nous remarquons comme attendue, une diminution du nombre de caractères entre le texte originale et le texte abrégé, correspondant à des pourcentage de réduction de 20 à 30%. De plus nous observons des temps d'exécution acceptable pour une utilisation sur Internet. Nous pouvons remarquer que le temps d'exécution n'est pas directement lié au nombre de mots du texte initial. De plus, il n'est pas non plus possible

	Texte original			Texte abrégé		
	Nb de mots	Nb de caractères	Nb moyen de caractères par mots	Nb de caractères	Pourcentage de réduction	Temps d'exécution en ms
$Texte1$	1077	2466	2, 28	1828	25%	172ms
$Texte2$	1346	2960	2, 19	2181	26%	391ms
$Texte3$	5231	12877	2, 46	9147	28%	703ms
$Texte4$	1721	3749	2, 17	2697	28%	453ms
$Texte5$	679	1808	2, 66	1250	30%	282ms
$Texte6$	869	2240	2, 57	1659	25%	297ms
$Texte7$	7870	19963	2, 53	14297	28%	2188ms
$Texte8$	2019	4930	2, 44	3573	27%	546ms
$Texte9$	3817	9846	2, 57	7238	26%	1031ms
$Texte10$	852	2414	2, 83	1815	24%	390ms

TAB. 4.2 – Résultats de tests de conversion en braille abrégé sur différents textes.

d'établir de relation directe entre le genre du texte (pièces de théâtre pour les textes 1 et 2, contes pour enfants pour les textes 3 et 4, autobiographies pour les textes 5 et 8, ou encore romans pour les textes 6, 7, 9 et 10) et le temps d'exécution. Cependant — sauf quelques permutations — on peut fortement corréler le nombre de caractères au temps d'exécution. Il est à noter que, dans ce tableau, le nombre de mots indiqué ne représente pas les mots tels qu'ils sont définis dans la langue française, mais plutôt tels qu'ils sont utilisés dans le code de cette application. Ainsi les caractères de ponctuation et les espaces représentent des mots à part entière, ce qui explique que le nombre moyen de caractères par mot soit peu élevé. Les tests ont été effectués sur des textes relativement courts puisque cette application est destinée à transcrire des pages (ou des morceaux de pages) web, et non des documents importants comme des livres.

4.4 Optimisation d'un clavier virtuel par des fourmis artificielles

Dans la section 2.3 (page 69), différents systèmes de saisi de texte ont été présentés, notamment des méthodes de réorganisation des touches du clavier afin de minimiser la fatigue musculaire pour les utilisateurs d'un système de pointage.

Eggers et al. [56] ont eux aussi établi un algorithme pour créer un clavier optimal pour une langue donnée. L'originalité de leur travail réside dans l'utilisation de fourmis artificielles. Le principe de cet algorithme a été présenté dans la section 2.3 et le rôle des fourmis artificielles a été détaillé dans la section 3.4.3 page 95. Ils ont ainsi montré que les fourmis artificielles peuvent trouver une organisation optimale des touches d'un clavier matériel ergonomique. Cependant, ce travail comporte deux principales limites : (1) une fois le clavier optimum trouvé pour une langue donnée, le programme d'optimisation n'a plus lieu d'être lancé de nouveau (sauf pour concevoir un clavier pour une autre langue ou pour ajouter de nouvelles touches), et, (2) même si les claviers actuels ne sont pas optimums, une nouvelle organisation des touches, même meilleure, ne sera pas adoptée par les utilisateurs (ce constat a été établi, pour tous les claviers visant à améliorer la vitesse de saisie et le confort par réorganisation des touches, dans la section 2.3.1).

Dès lors, avec l'adoption des claviers virtuels par les personnes handicapées, nous avons pensé que les problèmes de la méthode de Eggers and al. pouvaient être dépassés. Les déficiences sont en effet très différentes d'une personne à l'autre, ce qui introduit l'idée de générer des claviers différents pour chaque utilisateur. Cette idée s'impose davantage puisqu'une personne handicapée sera plus motivée pour utiliser un clavier personnalisé, adapté au type de texte qu'elle désire saisir.

4.4.1 Modélisation et algorithme

Ce problème peut être formulé comme suit :
- un clavier de $m \times n$ touches
- un ensemble de textes à saisir, ou un ensemble de documents produits généralement par l'utilisateur dans un contexte précis (financier, programmation, langage courant par exemple dans les emails...)

En somme, le but est de trouver la meilleure organisation du clavier pour minimiser les mouvements de l'utilisateur.

Le rôle des fourmis va donc consister à affecter à chaque emplacement vide du clavier un nouveau symbole (c'est-à-dire une lettre ou une touche). Pour cela, elles sont influencées par la quantité de phéromones déposées par leurs congénères. Différentes possibilités d'utilisation des phéromones sont envisageables :

1. les phéromones sont utilisées par les fourmis pour choisir la position appropriée sur le clavier en construction de chaque symbole,

2. ou bien, les phéromones sont utilisées par les fourmis pour choisir le prochain symbole à assigner à chaque étape de la construction de la solution.

Eggers et al. [56] utilisent dans leur méthode la première possibilité. Ils proposent en effet un placement optimum des touches sur un clavier ergonomique séparant les touches en deux zones destinées à être utilisées par chacune des mains. De plus ils évaluent ce placement selon le positionnement des touches les plus fréquemment utilisées sous la main (pour favoriser l'utilisation de l'index et du majeur), et selon l'alternance des deux mains pour la saisie de caractères consécutifs. De notre côté, nous avons porté notre attention sur l'ordre des touches — pour réduire les déplacements lors de la saisie avec un stylet — et non sur leurs positions sur le clavier — laquelle se révèle plus importante lors de la saisie à deux mains. Par la suite, nous allons donc considérer la deuxième possibilité : les phéromones seront de ce fait déposées entre les symboles et les fourmis se concentreront sur l'ordre des symboles.

L'algorithme 3 décrit la méthode de génération d'un clavier virtuel, c'est-à-dire d'affectation d'un symbole à chaque emplacement vide du clavier. Il correspond à un algorithme classique de fourmis artificielles : les fourmis construisent des solutions, dont la meilleure est utilisée pour mettre à jour la mémoire globale ; et ce pour un nombre donné d'itérations.

Algorithme 3 Optimisation de l'organisation des touches d'un clavier avec des fourmis artificielles

```
1: Initialiser les valeurs des phéromones τ⁰_{i,j}
2: Pour T^max itérations Faire
3:     Pour toute fourmi k
4:     Construire une organisation du clavier K_k
5:     Evaluer la qualité de K_k
6:     Fin Pour
7:     Mettre à jour la valeur des phéromones selon la qualité de la nouvelle solution et
       l'évaporation naturelle des phéromones
8: Fin pour
9: Retourner le meilleur clavier trouvé depuis le début
```

4.4.2 Construction de la solution

A chaque itération $T \in \{1, \ldots, T^{\max}\}$, chaque fourmi k construit une solution (c'est-à-dire un clavier) $\mathcal{K}_k$ en s'appuyant sur une mémoire collective — appelée phéromone dans leur milieu naturel. Les phéromones artificielles sont des valeurs réelles qui sont utilisées par les fourmis pour construire une solution. Notons $\tau_{i,j}$, la quantité de phéromones entre le symbole i et le symbole j. Ajoutons que les fourmis se servent également d'une information locale $\eta_{i,j}$ appelée « désirabilité ». Cette dernière est calculée au début de l'exécution du programme et évolue au cours des itérations.

Au temps t de l'itération T, la fourmi a construit la solution partielle $\mathcal{K}_k(t)$, où i est le dernier symbole assigné. La probabilité d'affecter le symbole j à la suite de i dans ce clavier

en construction est donnée par :

$$P^{(t)}(i,j) = P_e^{(T)} \times \frac{\tau_{i,j}^{\alpha} \times \eta_{i,j}}{\displaystyle\sum_{l \in N(\mathcal{K}_k(t))} \tau_{i,l}^{\alpha} \times \eta_{i,l}} + (1 - P_e^{(T)}) \times \begin{cases} 1 & \text{si } j = \arg \max\limits_{l \in N(\mathcal{K}_k(t))} \{\tau_{i,l}^{\alpha} \times \eta_{i,l}\} \\ 0 & \text{sinon} \end{cases} \tag{4.1}$$

où j est choisi dans l'ensemble $N(\mathcal{K}_k(t))$, lequel correspond à l'ensemble des symboles restant à affecter dans le clavier en cours de construction $\mathcal{K}_k(t)$. L'exposant α correspond à un paramètre déterminant l'importance des phéromones par rapport à la désirabilité. $P_e^{(T)}$ est la probabilité d'exploration/exploitation, laquelle est souvent constante dans les algorithmes de fourmis. Dans notre algorithme, cette valeur varie au cours des itérations de la manière suivante : $P_e^{(T)} = 0.8(T/T^{\max})$. Cette variante a pour but d'augmenter le comportement d'exploitation des fourmis par rapport à l'exploration lorsque les itérations augmentent.

Au temps $t = 0$, la fourmi commence avec un symbole virtuel utilisé uniquement comme point de départ. Au temps $t = m \times n$, la fourmi a terminé de construire sa solution et le clavier obtenu peut être évalué. La figure 4.11 montre un exemple d'organisation des touches d'un clavier.

	0	1	2	3	4	5	6	7	8	9	10	11
0	(	a	z	e	r	t	y	u	x	b	:	)
1	!	i	o	p	q	s	d	f	c	n	.	;
2	*	,	g	h	j	k	l	m	w	v	?	

FIG. 4.11 – Clavier de taille 12×3.

4.4.3 Evaluation de la solution

La qualité $Q(\mathcal{K}, S)$ d'un arrangement $\mathcal{K}$ est calculée selon une séquence de symboles $S = \{s[1], s[2], \ldots, s[|S|]\}$ de longueur $|S|$. La qualité correspond à la longueur des mouvements nécessaires pour saisir la séquence sur le clavier $\mathcal{K}$.

$$Q(\mathcal{K}, S) = \frac{100}{|S|} \sum_{i=1}^{|S|-1} d_{\varepsilon}(s[i], s[i+1]) \tag{4.2}$$

où $d_{\varepsilon}(x, y)$ correspond à la distance euclidienne entre les touches x et y de coordonnées $(x_1, x_2) \in \{0, \ldots, m-1\}^2$ et $(y_1, y_2) \in \{0, \ldots, n-1\}^2$ sur le clavier $\mathcal{K}$:

$$d_{\varepsilon}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{4.3}$$

Ainsi, plus $Q(\mathcal{K}, S)$ est faible, meilleure est la qualité du clavier $\mathcal{K}$ pour le texte S.

4.4.4 Mise à jour des phéromones

Dès que toutes les fourmis ont construit leur solution, la valeur de chaque phéromone est mise à jour selon la règle suivante :

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} + \rho \sum_{k=1}^{A} \Delta_{i,j}^{k} \tag{4.4}$$

Dans cette équation, ρ est appelé coefficient d'évaporation et Δ est calculé pour favoriser un bon choix pour les fourmis :

$$\Delta_{i,j}^{k} = \begin{cases} \frac{\min\limits_{l \in \{1,\ldots,A\}} \{Q(\mathcal{K}_l, S)\}}{Q(\mathcal{K}_k,S) \times (\text{Rank}(\mathcal{K}_k)+1)} & \text{si le symbole } j \text{ suit le symbole } i \text{ dans le clavier } \mathcal{K}_k \\ 0 & \text{sinon} \end{cases} \tag{4.5}$$

où A est le nombre de fourmis.

En définitive $\Delta_{i,j}^{k}$ augmente à chaque fois qu'une fourmi place dans son clavier la touche j après la touche i, et $\Delta_{i,j}^{k}$ est inversement proportionnel à la qualité du clavier ainsi qu'à son rang dans l'ordre décroissant des qualités ($\text{Rank}(\mathcal{K}_{\text{best}}) = 1$). Donc $\Delta_{i,j}^{k}$ augmente d'autant plus lorsque la qualité du clavier est bonne (c'est-à-dire lorsque $Q(\mathcal{K}, S)$ est petit) et lorsque meilleur est ce clavier comparé aux autres claviers générés. Par ailleurs, si aucune fourmi n'a décidé de placer le symbole j après le symbole i, la quantité de phéromones entre ces deux symboles s'évapore naturellement selon le paramètre ρ. Finalement la valeur des phéromones est contenue dans l'intervalle $[\tau_{\min}; \tau_{\max}]$.

4.4.5 Résultats expérimentaux

Dans le but d'étudier les performances de notre algorithme, nous avons utilisé un ensemble de 12 documents différents classés en 4 types (table 4.3). A chaque fois on utilise un clavier de taille 20×3 ainsi que le même ensemble de caractères, à savoir :

```
azertyuiopmlkjhgfdsqwxcvbn,;:!.?&"'(-_)][$* 1234567890{}\n
```

Nom	Type	Longueurs
S_1, S_2, S_3	extraits de blog	$1564, 3269, 2370$
S_4, S_5, S_6	codes de programmation en langage C	$1943, 3495, 2934$
S_7, S_8, S_9	extraits des Contes de Grimm	$8802, 9425, 6193$
S_{10}, S_{11}, S_{12}	extraits de journaux	$4103, 2105, 4701$

TAB. 4.3 – Les documents utilisés pour les tests.

Les paramètres du programme ont été choisis comme suit :
- intervalle des valeurs de phéromones : $[\tau_{\min}; \tau_{\max}] = [0.1; 0.9]$,

- valeur initiale des phéromones : $\tau_{i,j}^0 = \tau_{\max}$ si $i \neq j$ et $\tau_{i,i}^0 = 0.0$,
- coefficient d'évaporation : $\rho = 0.01$,
- désirabilité $\eta_{i,j}$: calculée selon la fréquence relative de co-occurence des symboles i et j dans le texte considéré,
- nombre de fourmis = nombre de symboles à placer sur le clavier (57 dans notre cas),
- nombre d'itérations : 500 (chaque fourmi construit 500 claviers).

La figures 4.14, 4.13 et 4.12 montrent respectivement l'évolution de la qualité des claviers, de la valeur des phéromones et celle de l'entropie des phéromones (équation 4.6) pour le document S_1 (blog).

$$\mathrm{Ent}(T) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|\tau_{i,j} - \bar{\tau}|}{\max\limits_{u,v}\{\tau_{u,v}\} - \min\limits_{u,v}\{\tau_{u,v}\}} \log \frac{|\tau_{i,j} - \bar{\tau}|}{\max\limits_{u,v}\{\tau_{u,v}\} - \min\limits_{u,v}\{\tau_{u,v}\}} \tag{4.6}$$

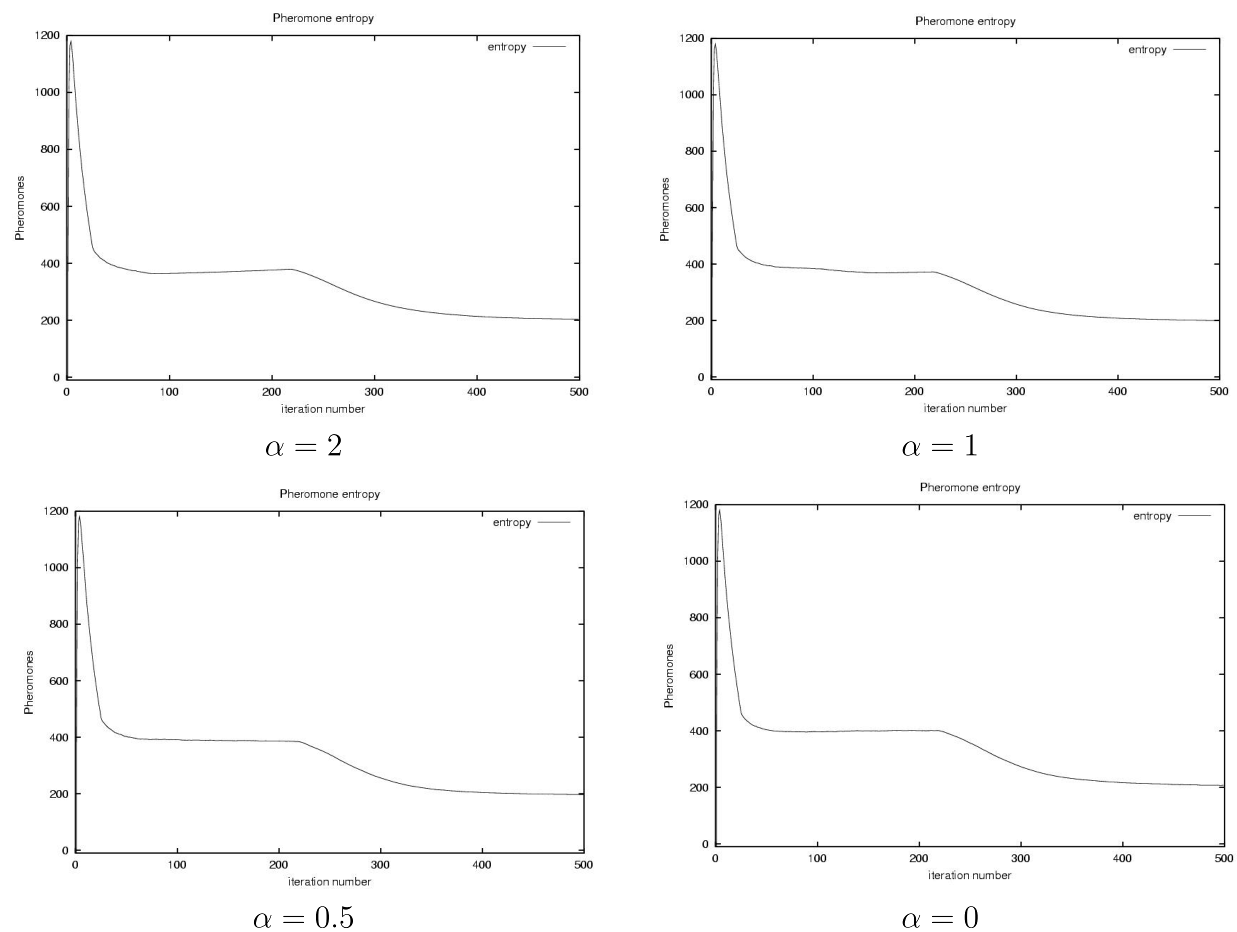


FIG. 4.12 – Entropie des phéromones pour le document S_1 (blog) avec différentes valeurs de α.

Quatre valeurs de α sont comparées : $\alpha = 1$ (les phéromones et la désirabilité ont la même importance dans la formule de probabilité de décision des fourmis), $\alpha = 2$ (les phéromones sont deux fois plus utilisées que la désirabilité), $\alpha = 0.5$ (les phéromones sont deux fois moins utilisées que la désirabilité), et $\alpha = 0$ (les phéromones ne sont pas utilisées). Chaque courbe représente une moyenne de 30 exécutions indépendantes. Les courbes obtenues pour les autres documents sont très similaires à celles de ce document.

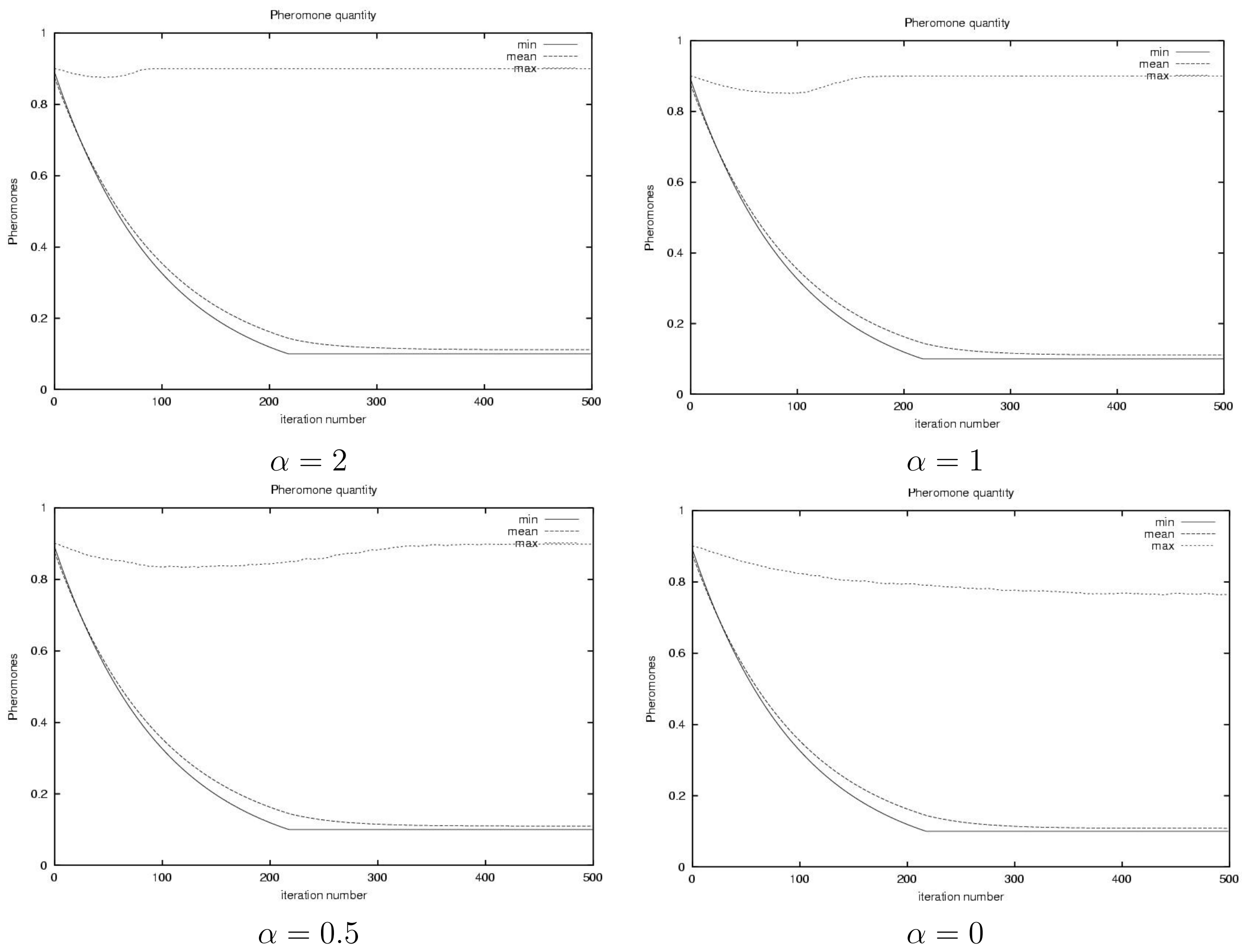
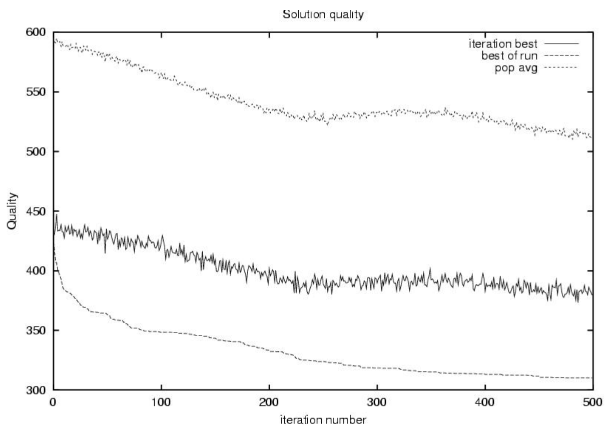


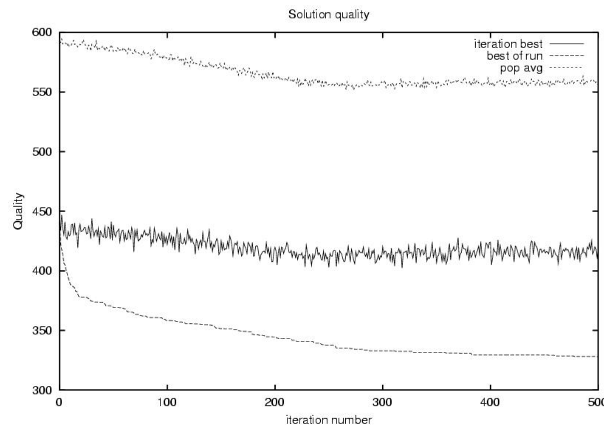
FIG. 4.13 – Valeur des phéromones (max, min et moyenne sur tous les chemins) pour le document S_1 (blog) avec différentes valeurs de α.

Nous pouvons remarquer que la courbe d'entropie des phéromones (figure 4.12) est plus ou moins identique quelle que soit la valeur de α : l'entropie des phéromones dépend uniquement de la valeur moyenne des phéromones, laquelle décroit de manière identique dans tous les cas (lorsque la plupart des chemins ont atteint la valeur minimale de phéromones $\tau_{\min}$ aux alentours de l'itération $T = 220$, voir figure 4.13). Cependant, on peut voir que lorsque les phéromones ne sont pas utilisées (figure 4.14 avec $\alpha = 0$), la qualité moyenne de la population et le meilleur clavier pour une itération donnée restent constants alors que pour $\alpha = 2.0$ la qualité des claviers générés par l'ensemble de la population diminue (avec

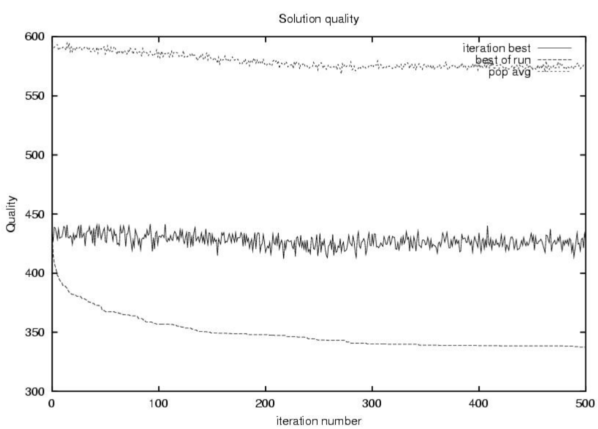
un ralentissement autour de l'itération $T = 220$) ; ce qui signifie que les claviers sont de plus en plus adaptés au type de texte à saisir.



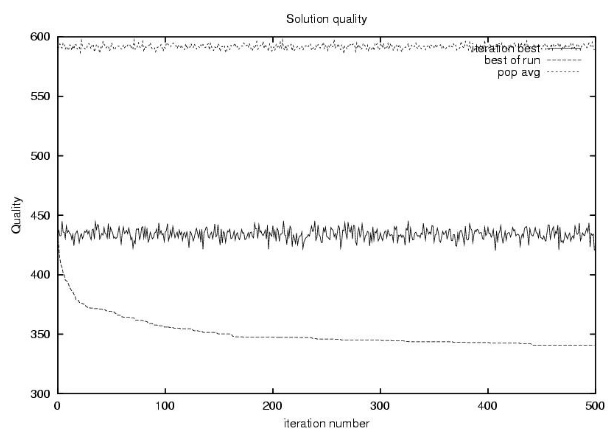
$\alpha = 2$



$\alpha = 1$



$\alpha = 0.5$



$\alpha = 0$

FIG. 4.14 – Evolution de la qualité du clavier (le meilleur depuis le début de l'exécution, le meilleur de l'itération, la moyenne sur toute la population) pour le document S_1 (blog) avec différentes valeurs de α.

Pour chaque document $\{S_1, \ldots, S_{12}\}$ et pour chacune des 30 exécutions indépendantes, nous avons obtenu un clavier virtuel. Ce clavier est alors utilisé lors de la saisie des 12 documents pour évaluer sa qualité. Les valeurs reportées dans le tableau 4.4 donnent la performance d'un clavier par rapport aux 12 documents. Nous pouvons remarquer que pour un clavier donné — c'est-à-dire généré pour un certain document —, la meilleure performance lors de la saisie d'un texte est obtenue lorsque ce dernier a été utilisé pour générer le clavier, et plus généralement, lorsqu'il est du même type que le texte de référence. Cela nous confirme qu'il est plus performant d'utiliser un clavier adapté à l'activité de l'utilisateur. Notamment, force est de constater que les claviers construits à partir de texte en langage C sont réellement plus performants que les autres pour saisir du code en C.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
S_1	1.89	5.43	1.47	37.24	73.75	29.69	1.90	4.30	2.90	3.40	3.19	3.50
S_2	9.84	1.66	0.73	44.49	80.19	29.87	3.68	5.37	5.78	6.17	6.77	4.53
S_3	11.77	7.29	0.20	39.42	82.88	26.97	6.07	8.82	7.23	7.30	7.75	7.28
S_4	21.57	19.86	15.13	1.09	42.07	10.46	17.49	20.04	18.21	17.28	16.87	17.94
S_5	50.35	47.34	45.06	38.04	0.00	33.10	47.63	46.64	42.21	49.04	50.38	47.10
S_6	74.34	70.69	69.15	90.06	109.08	0.00	73.23	72.12	70.19	71.02	69.74	70.33
S_7	8.39	7.65	4.12	41.56	74.31	30.33	0.25	4.31	3.95	5.27	6.59	5.29
S_8	7.83	5.26	1.80	38.73	75.70	25.43	1.48	2.53	2.53	5.21	5.22	4.74
S_9	8.35	8.04	4.50	41.61	76.80	31.86	2.18	5.31	0.60	7.53	7.15	5.80
S_{10}	7.24	5.94	1.64	36.87	74.87	25.55	2.85	5.07	4.10	1.29	4.01	2.84
S_{11}	6.81	7.32	2.68	41.08	78.09	25.35	3.01	5.69	4.79	3.85	1.27	3.72
S_{12}	6.16	7.46	2.39	42.00	87.65	31.73	2.51	5.56	4.52	3.91	5.07	1.41

TAB. 4.4 – Performance du meilleur clavier généré pour chaque document (en ligne) pour saisir les 12 documents (en colonne) de types différents. Les résultats sont donnés en %.

Afin d'évaluer les performances de notre algorithme de fourmis artificielles, nous l'avons comparé à deux autres méthodes de recherche locale, à savoir la Recherche Taboue [66] et le Recuit Simulé [70][124]. Le tableau 4.5 représente la qualité moyenne (sur 30 exécutions indépendantes) ainsi que l'écart moyen, des claviers obtenus pour chacun des 12 documents, avec notre méthode de Fourmis Artificielles (FA), avec la Recherche Taboue (RT), et avec le Recuit Simulé (RS). Par exemple, en moyenne la qualité des claviers obtenus pour la méthode des fourmis est 377,72% pour le document S_1 avec un écart moyen (la moyenne de la valeur absolue des écarts à la moyenne) de 10,01.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
FA	377.72 (10.01)	364.69 (9.92)	356.76 (10.10)	406.33 (8.92)	432.69 (9.72)	338.37 (9.51)	363.78 (11.29)	370.93 (8.20)	360.69 (10.55)	367.62 (12.48)	367.03 (12.32)	367.92 (9.44)
RT	551.55 (53.04)	580.15 (94.18)	529.03 (61.02)	530.84 (61.79)	555.92 (67.38)	536.84 (75.31)	540.52 (79.55)	566.01 (75.19)	521.90 (68.62)	549.64 (73.98)	518.88 (86.39)	519.78 (59.65)
RS	435.74 (29.98)	435.94 (30.81)	422.31 (30.66)	465.53 (21.49)	491.78 (23.07)	414.02 (31.43)	450.32 (29.96)	442.76 (22.84)	428.74 (25.97)	443.43 (28.92)	456.38 (27.60)	453.32 (21.95)
FA + RL	371.78 (10.61)	362.44 (7.23)	349.85 (11.60)	403.46 (10.03)	430.59 (11.40)	333.38 (8.60)	360.94 (10.12)	364.20 (10.87)	352.76 (9.97)	364.96 (10.63)	368.92 (10.31)	363.58 (11.62)

TAB. 4.5 – Qualité des claviers générés pour chaque document avec les méthodes des Fourmis Artificielles (FA), de la Recherche Taboue (RT), du Recuit Simulé (RS), et des Fourmis Artificielles couplée avec une Recherche Locale (FA+RL). Les résultats sont donnés en %.

Nous observons de meilleurs résultats avec la méthode des fourmis artificielles qu'avec celles des recherches locales. Notons que les deux méthodes de recherches locales nécessitent à l'initialisation une solution réalisable, c'est-à-dire une organisation des touches sur le clavier, qui est améliorée au fur et à mesure des itérations. Ce clavier initial a été généré de manière aléatoire et est donc très différent d'une exécution à une autre. Ainsi sur les 30 exécutions indépendantes, les méthodes de recherche locale sont initialisées avec des claviers dont la qualité peut être très bonne comme très mauvaise, ce qui explique les écarts moyens importants. Afin d'améliorer les performances de notre algorithme, nous lui avons associé une méthode de Recherche Locale (ligne FA+RL du tableau 4.5). Cette dernière est appliquée à chaque fois qu'une fourmi a généré un clavier, c'est-à-dire à des claviers qui s'améliorent au fur et à mesure des itérations, et les résultats obtenus influencent par la suite les autres fourmis de la colonie. La recherche locale employée ici est la méthode FirstFit

qui consiste à effectuer une permutation au hasard et à la conserver si elle constitue une amélioration. Cette recherche rapide permet d'améliorer légèrement la qualité des claviers en gardant un temps d'exécution acceptable pour l'utilisateur (le tableau 4.6 représente le temps moyen d'exécution sans puis avec la recherche locale).

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
FA	30.71	41.61	35.63	32.06	38.63	35.11	74.57	79.29	58.82	45.68	33.80	49.99
FA+RL	45.07	64.72	53.38	47.99	60.89	53.54	136.40	139.37	99.27	73.13	50.22	80.73

Tab. 4.6 – Durée de génération moyenne (sur les 30 exécutions) des claviers pour chaque document avec la méthode des Fourmis Artificielles sans Recherche Locale (FA) puis avec (FA+RL). Les résultats sont donnés en secondes.

4.5 Conclusion

Dans ce chapitre, nous avons exposé différents dispositifs d'aide, que ce soit pour le webmestre (avec l'outil d'aide à l'accessibilité des sites web), ou pour l'internaute (avec un générateur de braille abrégé ou bien un clavier virtuel pour personnes handicapées).

D'une part, notre outil d'aide propose à un webmestre débutant une démarche personnalisée pour créer ou rendre un site web accessible. Tout au long de celle-ci, il a la possibilité d'utiliser divers outils comme des fiches techniques, des aides à la correction des erreurs et un validateur des normes d'accessibilité. Rappelons que ce dernier est basé sur le validateur OCAWA, dont nous sommes de ce fait dépendants pour garantir le fonctionnement de notre validateur. Pour remédier à ce problème, nous avons décidé d'en créer un qui ne dépende pas d'un validateur commercial, tout en reprenant le principe d'affichage des erreurs par thème présenté dans ce chapitre. Ainsi la réalisation de ce premier validateur nous a permis d'envisager la réalisation d'un validateur plus complet (c'est-à-dire capable de tester diverses normes) et facilement utilisable par des webmestres débutants. Ce validateur sera présenté dans la section 5.2 (page 127). Finalement, cet outil d'aide nous a permis de nous familiariser avec les normes d'accessibilité puisqu'il a été créé sous la forme d'un site web accessible (consultable à l'adresse `http://accessi.hant.li.univ-tours.fr/aideaccessi/accueil.php`), et nous a également permis de réfléchir sur une démarche méthodologique de migration compréhensible par des webmestres débutants.

D'autre part, nous avons présenté des systèmes d'aide pour accélérer non seulement la vitesse de lecture en braille mais aussi celle de la saisie sur un clavier virtuel. Tout d'abord, nous avons présenté le générateur de braille abrégé pour lequel nous envisageons deux types d'utilisation sous la forme de plugin : pour navigateur web, ou bien pour un système permettant de lire plus rapidement les pages web avec un lecteur d'écran — comme celui que nous présenterons dans la section 6.3. Ensuite, notre algorithme d'organisation des touches sur un clavier virtuel par des fourmis artificielles permet à des personnes handicapées de réduire leurs mouvements pour saisir un type de texte donné. Remarquons que ce problème de placement des touches sur un clavier (KAP, Keyboard Arrangmnent Problem) peut également être perçu comme une application d'un problème d'affectation quadratique (QAP, Quadratic Assignment Problem) [103]. De ce fait, des méthodes de résolution du QAP devront être implémentées et comparées à notre méthode. Etant donné l'application concrète, nous avons intégré notre programme d'organisation des touches au clavier CVK présenté dans la section 2.3.2 pour pouvoir par la suite être testé dans des conditions réelles. Plusieurs profils, ou, dans notre cas, des organisations de touches pour un type de texte, peuvent ainsi être sauvegardés sur le clavier virtuel.

Les travaux présentés dans ce chapitre ont donné lieu à l'encadrement de plusieurs projets de fin d'études au Département d'Informatique (DI) de Polytech'Tours [61][77][4] et d'un stage de type analyste-programmeur [65]. Ils ont également conduit à un certain nombre de communications ou de publications [35][39][34][36].

Chapitre 5

Outils de synthèse

Nous présenterons dans ce chapitre deux outils de synthèse que nous avons réalisés. Le premier est un validateur des normes d'accessibilité pour tester les pages d'un site web. Il est multi-pages et multi-recommandations. Le second s'adresse aussi bien aux webmestres qu'aux internautes ; il s'agit d'un générateur de plans de sites web pouvant s'adapter aux capacités des personnes non-voyantes. Pour ce faire nous avons eu recours aux fourmis artificielles.

5.1 Introduction

Le manque de recul des webmestres face à l'accessibilité constitue un frein important : vouloir aller toujours plus vite en matière d'administration de site web n'est pas forcément la bonne méthode. En revanche, prendre le temps d'étudier, d'analyser la situation et de synthétiser le tout, peut considérablement améliorer la vitesse et la qualité des tâches futures.

D'une part, lors de la mise en conformité de son site web, le webmestre dispose de nombreux outils de détection des erreurs. Comme nous l'avons présenté dans la section 1.4.3, chacun est capable de tester certaines recommandations et possède sa propre manière de présenter les résultats. La prise en main de tous ces validateurs différents pour le webmestre débutant est longue et décourageante. Nous avons donc créé un validateur unique, qui sera présenté dans la section 5.2 de ce chapitre, permettant de tester les différentes normes requises pour obtenir un site web accessible.

D'autre part, l'ajout anarchique de nouvelles pages dans un site web conduit rapidement à le rendre inutilisable. La navigation au sein du site devient en effet difficile voire impos-

sible. Pour éviter de se perdre dans les nombreuses pages web nous avons étudié, dans la section 2.2, la possibilité d'utiliser les plans de site. Le webmestre peut ainsi avoir une vue globale de son site avant d'y ajouter de nouvelles pages, en utilisant un générateur de plan. Il en est de même pour un internaute désorienté. Toutefois les générateurs étudiés ne font que reproduire, sous différentes formes, la structure réelle du site. Par conséquent lorsque l'internaute se trouve face à un site web mal structuré, la navigation peut s'avérer autant difficile avec que sans plan de site, surtout si ce dernier est généré automatiquement. Nous avons donc créé un générateur de plan de site capable de fournir un plan regroupant les pages similaires et s'adaptant aux capacités de mémorisation des internautes non-voyants. Cet outil, qui utilise des fourmis artificielles pour générer les plans, sera présenté dans la section 5.3, dans laquelle nous montrerons l'intérêt de ce type de générateur, tant pour l'internaute que pour le webmestre.

5.2 Validateur multi-recommandations

Des validateurs existent (section 1.4.3 page 40) pour tester automatiquement une page web. Ils permettent de vérifier l'adéquation du code source de la page web avec les normes HTML, CSS ou encore WCAG. Ces validateurs, accessibles *via* Internet, sont utilisables gratuitement ou permettent — pour les validateurs commerciaux — de tester un nombre limité de pages web. Les différents validateurs étant indépendants, chacun possède sa propre manière d'afficher les erreurs décelées sur la page web testée. Rappelons que lorsqu'un webmestre souhaite rendre accessible une page web, il doit tester le code HTML puis le code CSS et enfin tester l'accessibilité. Il doit de ce fait utiliser trois validateurs différents et s'habituer à chaque méthode d'affichage. Imaginons par exemple le travail d'adaptation d'un webmestre débutant se servant du validateur HTML [252] du W3C — lequel présente les erreurs sous forme de liste dans l'ordre d'apparition dans la page testée — puis utilisant le validateur « The Wave » [261] — lequel affiche les erreurs sous forme graphique dans une visualisation de la page web. Dans l'optique de faciliter cette démarche de validation automatique, nous avons créé un validateur multi-recommandations (figure 5.1). Il peut être utilisé seul pour tester la validité d'une page web, puisqu'il permet de tester, en une seule passe de validation, les normes HTML, CSS et celles d'accessibilité.

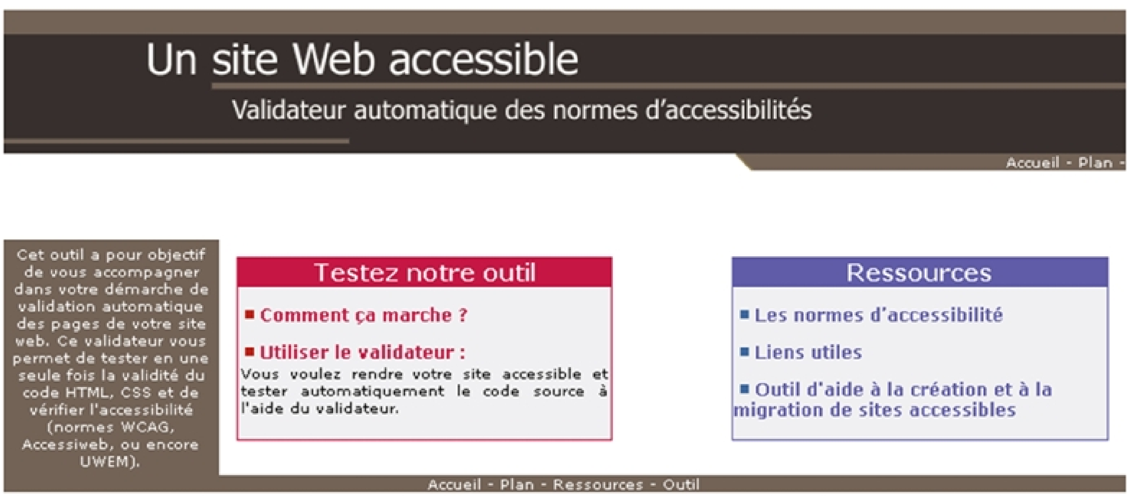


Fig. 5.1 – Page d'accueil de notre validateur multi-recommandations.

5.2.1 Présentation du validateur

Avant de commencer à analyser un site web, le webmestre doit bien comprendre les différentes étapes qui permettront de le rendre valide. Dans ce but, notre validateur intègre une partie « Ressources » disponible depuis la page d'accueil. Elle contient les définitions des normes qu'un site accessible se doit de respecter, c'est-à-dire les normes HTML, CSS et d'accessibilité. De plus, pour approfondir la notion d'accessibilité des sites web, une

liste des liens les plus importants dans ce domaine est rappelée. Finalement, cette partie « Ressources » présente une brève introduction à l'accessibilité. Toutefois, si le webmestre ne sait vraiment pas par où commencer, il peut en amont utiliser notre « outil d'aide à la création et à la migration de sites Internet accessibles » (présenté dans la section 4.2 page 103).

Dès que le webmestre se sent prêt à tester la validité de ses pages web, il peut se rendre dans la partie outil. Il a alors accès à quelques explications sur les possibilités offertes par le validateur, puis au validateur proprement dit.

En plus de tester en une seule passe les différentes recommandations à respecter pour rendre un site accessible, notre validateur possède aussi une fonctionnalité permettant de tester plusieurs pages web en une seule fois. Cela permet d'obtenir rapidement un aperçu du niveau d'accessibilité d'un site web et de repérer facilement les erreurs se répétant sur toutes les pages du site web. En effet, quelque soit la recommandation, un grand nombre d'erreurs n'est pas forcément impressionnant ni problématique si l'utilisateur se rend vite compte que son site ne possède en réalité que quelques erreurs se répétant plusieurs fois. Notre validateur affiche de ce fait les erreurs non pas par ordre d'apparition mais groupées par type d'erreur, cela pour une ou plusieurs pages. La figure 5.2 présente l'interface de lancement du validateur où l'on peut voir les différents champs à renseigner avant de lancer la validation multi-recommandations. Sur cette figure, on peut distinguer deux formulaires différents : celui de gauche permet de lancer la validation sur une ou plusieurs pages web, alors que celui de droite permet de visualiser les résultats d'une validation antérieure.

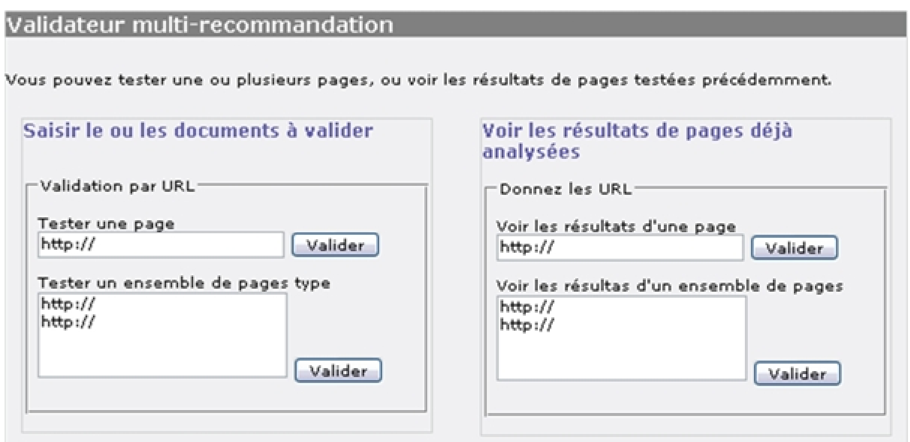


FIG. 5.2 – Interface de lancement du validateur multi-recommandations.

5.2.2 Fonctionnement et structure du validateur

Pour réaliser ce validateur multi-recommandations, nous avons choisi de réutiliser des validateurs open source existants. Une fois la validation lancée par l'utilisateur, les trois validateurs — à savoir HTML, CSS et WCAG — testent la ou les pages web. Les erreurs

décelées sont alors stockées dans une base de données de type MySQL. A ce moment, l'utilisateur peut choisir d'afficher les erreurs (voir section 5.2.4 page 132). La figure 5.3 illustre le principe de fonctionnement de notre validateur.

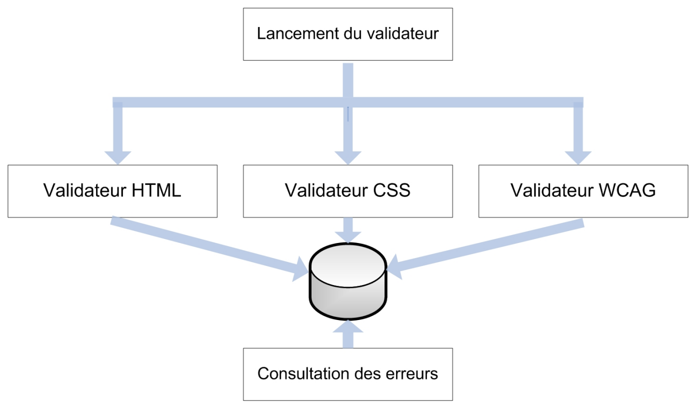


FIG. 5.3 – Principe de fonctionnement du validateur.

Choix des validateurs

Pour tester le code HTML et le code CSS, notre choix s'est porté sur les très reconnus validateurs du W3C. Afin de sélectionner le validateur d'accessibilité, nous avons utilisé le moteur de recherche d'outils de validation[1] du W3C. Nos critères ont été les suivants : logiciel gratuit, licence open source, test des normes WCAG, logiciel fournissant un rapport d'erreurs et utilisable en ligne. A la vue des résultats donnés par le moteur de recherche, nous avons choisi d'utiliser le validateur Hera, développé par Fundacio Sidar, qui satisfait tous nos critères.

La base de données

Nous souhaitons stocker dans la base de données tous les résultats d'analyse des sites web par les différents validateurs. Chaque page comporte éventuellement des erreurs, repérables dans le code source de la page web grâce à la ligne/colonne de début et la ligne/colonne de fin. Chaque erreur possède une importance (différenciation des erreurs et des warnings) et appartient à un type d'erreur spécifique : soit HTML, soit CSS, soit d'accessibilité. D'un côté, une erreur HTML ou CSS porte sur un ou plusieurs attributs appartenant à un thème (images, tableaux, couleurs, texte...). D'un autre coté, les erreurs d'accessibilité correspondent au non respect d'un point de contrôle d'une directive en particulier. A cela s'ajoute que chaque point de contrôle possède une priorité (1, 2, ou 3 comme indiqué dans les recommandations WCAG).

[1] http ://www.w3.org/WAI/ER/tools/advanced.php

Cette base de données est composée de deux parties : l'une statique et l'autre dynamique. Respectivement, on trouve dans l'une des données pérennes — telle la liste des points de contrôle des WCAG, leur appartenance à une directive, à un thème... — et dans l'autre, à savoir la partie dynamique, on retrouve des entités telles que les erreurs que comporte une page web, les pages web testées... Ces données dynamiques sont renseignées lors de la validation d'une ou plusieurs pages. La figure 5.4 représente le MCD de la base de données où l'on peut distinguer la partie statique et la partie dynamique.

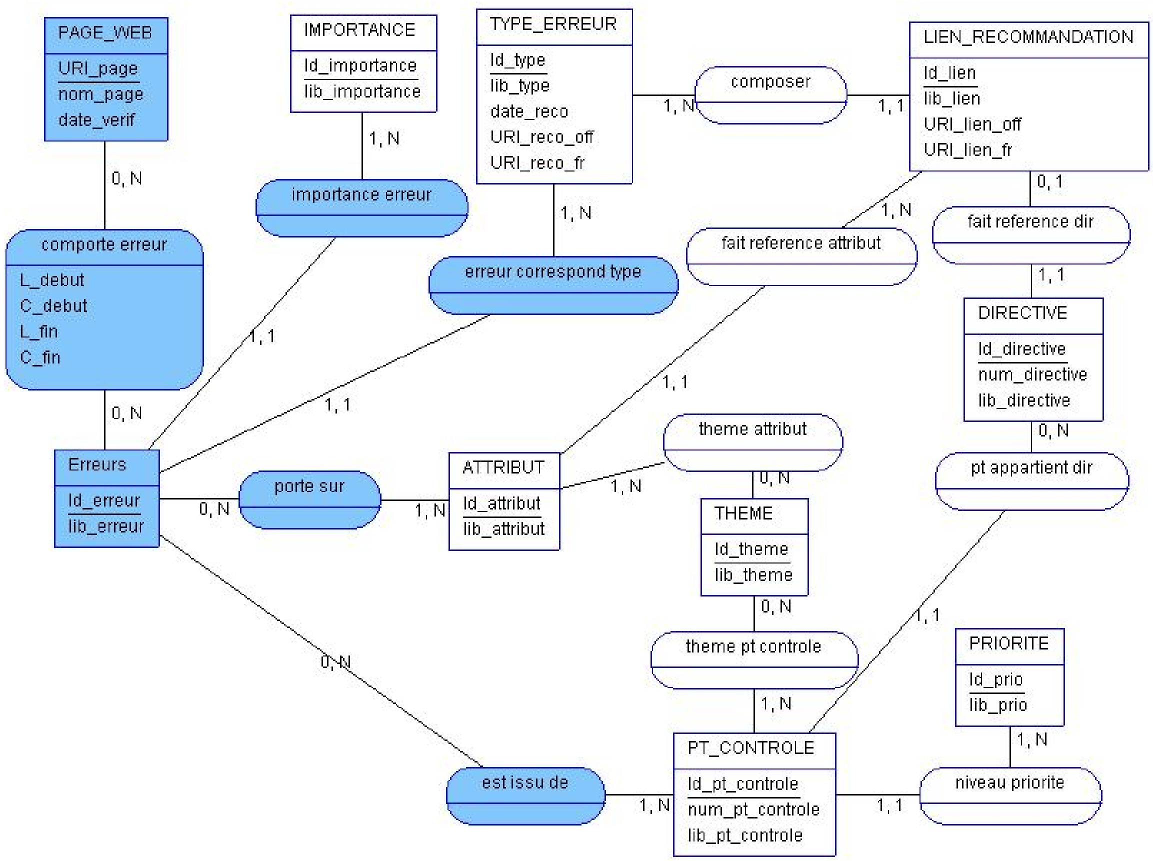


FIG. 5.4 – MCD comprenant des entités et relations statiques (fond blanc) et dynamiques (fond bleu).

5.2.3 Les normes supplémentaires

Notre validateur est multi-recommandations puisqu'il intègre les normes HTML, CSS et WCAG. Par ailleurs il inclut également les normes Accessiweb (label français) et UWEM (section 1.3.1 page 33). A l'aide du tableau de correspondance Accessiweb/WCAG, fourni par l'association BrailleNet, nous avons intégré les normes Accessiweb. Ainsi l'utilisateur peut avoir une idée du niveau Accessiweb que son site web peut atteindre.

Les normes UWEM ont été créées pour unifier la notion d'accessibilité du web en Europe. La France, avec l'association BrailleNet, a été le premier pays à participer à ce projet d'uniformisation en établissant un tableau de correspondance Accessiweb/UWEM. Actuellement, Accessiweb, Sello de Accessibilidad (Espagne) et, plus récemment (en mars 2007), AnySurfer (Belgique), ont rejoint ce projet Européen. Par la suite, ces normes pourront être intégrées à notre validateur en ajoutant dans la base de données une table contenant la liste des critères et des relations pour établir la correspondance avec les normes UWEM et WCAG.

A ce stade du développement, la base de données de notre validateur possède des tables supplémentaires correspondant aux points de contrôle Accessiweb et aux tests UWEM. La figure 5.5 illustre le MCD de la base de données du validateur. Précisons que la base de données peut facilement être mise à jour lorsque les normes évoluent.

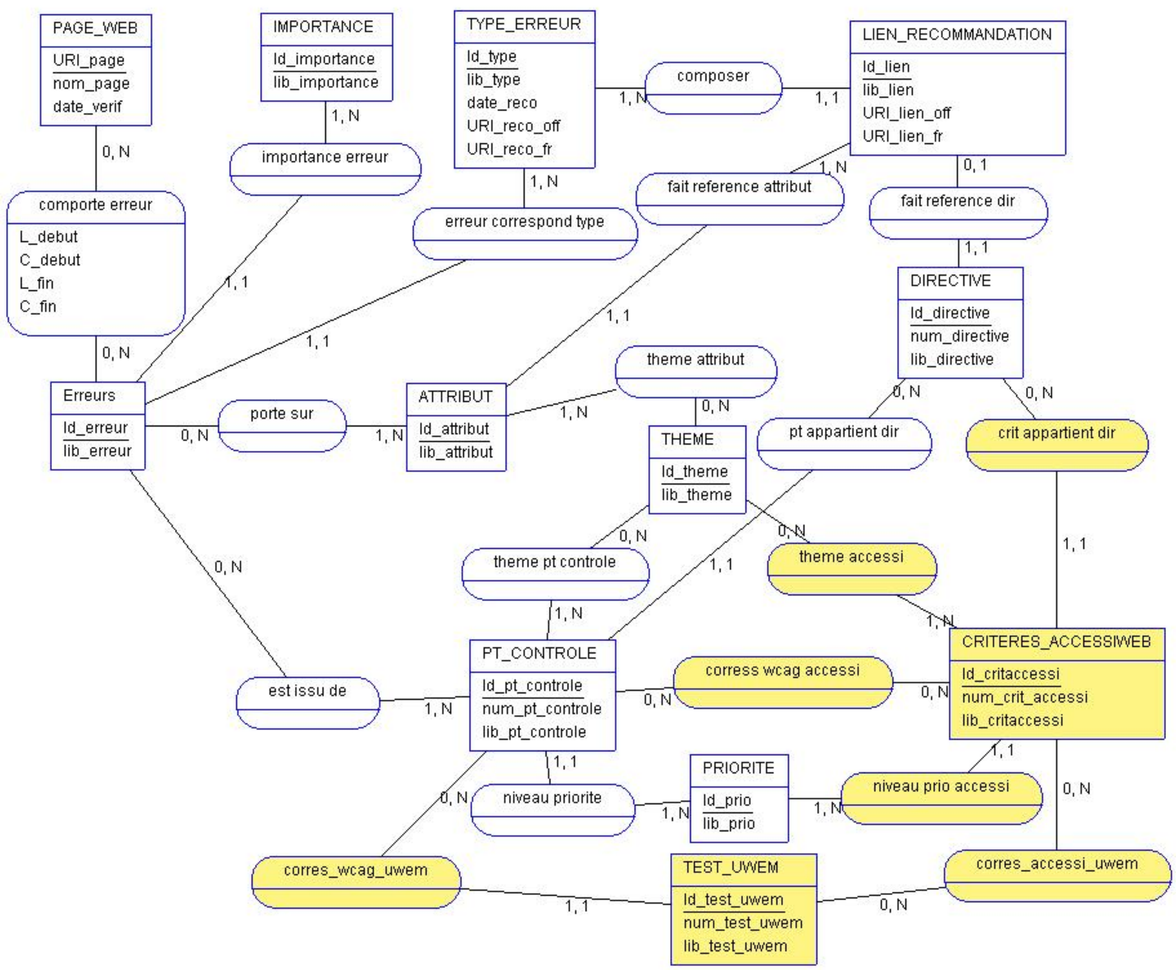


FIG. 5.5 – MCD comprenant les normes Accessiweb et UWEM (en jaune).

5.2.4 Affichage des résultats

Une fois la validation effectuée, tous les résultats sont stockés dans la base de données. L'affichage des erreurs peut se faire juste après l'analyse (comme pour les autres validateurs existant), ou plus tard lorsque l'utilisateur en fait la demande *via* la page de lancement du validateur (figure 5.2, formulaire de droite).

Choix des erreurs à afficher

A ce stade, l'utilisateur doit choisir le type d'erreur qu'il souhaite afficher. Les débutants préfèrent souvent faire des corrections étape par étape. Ils peuvent par exemple n'afficher dans un premier temps que les erreurs HTML, puis, une fois ces dernières corrigées, les erreurs CSS, et pour finir les erreurs d'accessibilité (WCAG ou Accessiweb). Les webmestres plus expérimentés peuvent quant à eux choisir d'afficher directement les erreurs HTML, CSS et d'accessibilité en une seule fois. Il est de plus possible de ne voir que les erreurs d'un thème particulier. Cette option est applicable avec n'importe quelle combinaison de choix de normes. La figure 5.6 illustre la page permettant de choisir les erreurs à afficher.



FIG. 5.6 – Interface permettant de choisir les erreurs à afficher.

Si l'utilisateur décide de visualiser les erreurs d'accessibilité, il pourra là encore procéder par étapes en commençant par les erreurs de priorité 1, 2, puis 3, ou afficher directement toutes les erreurs s'il souhaite atteindre le niveau le plus élevé d'accessibilité (niveau AAA des WCAG, ou niveau Or du label Accessiweb). Notons que l'affichage des erreurs est réalisé pour toutes les pages testées. Ainsi, lors d'une validation multi-pages, il est facile de repérer les pages comportant les mêmes erreurs.

Liste des erreurs

L'interface illustrée par la figure 5.6 informe également l'utilisateur du nombre d'erreurs que comportent la ou les pages qu'il a testé — soit, dans cet exemple, 27 erreurs HTML, 7 erreurs CSS et 46 erreurs WCAG. En reprenant le système d'affichage des erreurs sous forme de listing que l'on rencontre habituellement dans les validateurs, l'utilisateur aurait vu trois listing de 27 puis 7 puis 46 erreurs pour chacun des standards vérifiés. Or, bien souvent les mêmes erreurs se répètent plusieurs fois dans une page ou dans des parties redondantes de pages d'un même site. Dans cet exemple, les 46 erreurs WCAG relevées ne sont en fait que 11 erreurs différentes. Le travail répétitif de correction d'une même erreur n'est pas très difficile dès lors que l'utilisateur a compris comment y remédier. Ainsi, nous avons choisi d'afficher une entrée dans le listing pour chaque erreur différente, puis, pour chacune, d'indiquer le nombre de répétitions ainsi que les pages concernées. Cela permet de ne pas décourager le webmestre face à un nombre trop important d'erreurs.

La figure 5.7 représente l'affichage des erreurs HTML. Si l'erreur porte sur un attribut HTML existant alors celui-ci est spécifié et permet de faire un lien vers les recommandations HTML du W3C au chapitre traitant de cet attribut. Sur cette figure, la deuxième erreur se répète deux fois. Lorsque l'utilisateur décide d'en savoir plus concernant cette erreur, la (ou les) page(s) la contenant ainsi que la (ou les) ligne(s) concernée(s) apparait(ssent).

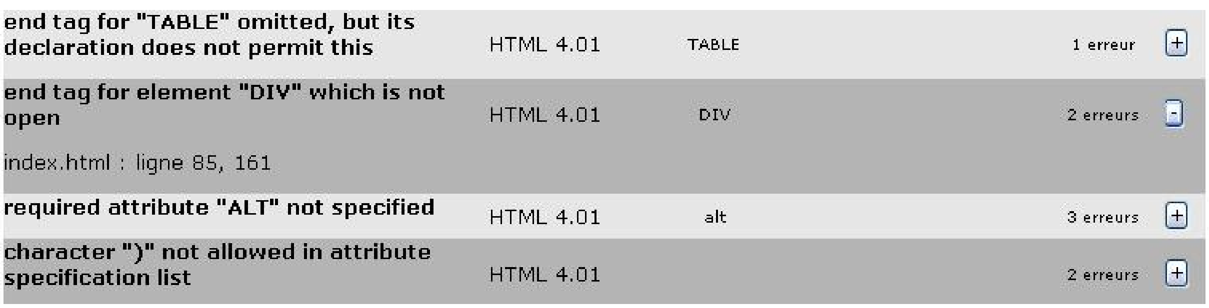


FIG. 5.7 – Affichage des erreurs HTML.

Concernant les erreurs CSS, l'affichage est identique à celui des erreurs HTML. En revanche, pour ce qui est des erreurs d'accessibilité, l'attribut n'est pas indiqué mais la directive des recommandations et la priorité de l'erreur le sont. A l'instar des attributs HTML et CSS, le numéro de la directive représente un lien vers les recommandations officielles au chapitre correspondant. Lorsque l'utilisateur ouvre l'entrée de l'erreur qui l'intéresse il voit le numéro du point de contrôle ainsi que son intitulé. Pour corriger cette erreur, il est possible de s'appuyer sur la liste des tests UWEM qui sont indiqués par la suite. Pour en savoir davantage, l'utilisateur peut se rendre sur le chapitre concerné dans les recommandations UWEM en cliquant sur le numéro du test. La figure 5.8 illustre un exemple d'erreur WCAG pour laquelle l'utilisateur a demandé le détail. Notons que l'affichage est similaire pour les erreurs Accessiweb.

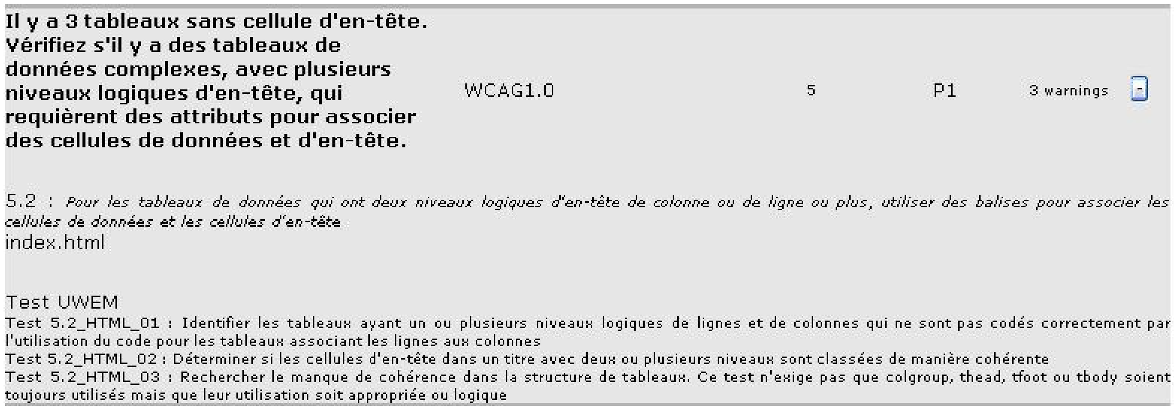


FIG. 5.8 – Affichage des erreurs WGAG.

Notre validateur permet ainsi à un webmestre débutant d'utiliser le même validateur pendant tout le travail de correction de son site web et de connaître la conformité de son site web au niveau européen.

5.3 Génération de plans de sites web par des fourmis artificielles

Un des problèmes majeurs d'Internet est de trouver l'information désirée. Pour naviguer plus facilement dans un site web jusqu'à la page pouvant contenir ce que cherche l'internaute, il peut utiliser le plan du site. Si ce dernier est absent, des générateurs existent (section 2.2 page 55). Toutefois, il ne s'agit là que de visualisations différentes pour représenter le plan du site tel qu'il existe. Un site offrant beaucoup de liens dès la page d'accueil aura également un plan complexe dès le premier niveau de l'arborescence représentant la structure du site. D'ailleurs, même avec un plan de site, la navigation peut être difficile — d'autant plus pour les personnes atteintes de handicaps. Arrêtons-nous sur le cas des personnes non-voyantes : si un plan de site leur permet de se repérer plus facilement, il reste tout de même difficile de mémoriser un nombre important de liens à chaque niveau de l'arborescence. Pour accéder à l'information désirée, il est souhaitable de n'avoir qu'un nombre limité de choix, correspondant ici au nombre de sous-pages.

Dès lors, nous proposons ici une méthode générant un plan de site indépendant de la structure réelle du site web mais adapté aux capacités de mémorisation de l'utilisateur. Notre approche permet à l'usager de préciser lui-même le nombre de sous-pages maximum auxquelles il désire avoir accès à chaque niveau de l'arborescence. Par conséquent, le plan généré doit respecter cette contrainte tout en proposant un arbre structuré selon les différentes thématiques abordées dans le site web.

La détermination par l'utilisateur du nombre de choix disponibles à chaque niveau hiérarchique en fonction de ses propres capacités assure de ne pas surcharger la représentation du plan. De ce fait le plan généré s'adapte aux difficultés rencontrées par les non-voyants lors de leur navigation sur Internet, puisqu'il leur permet de pouvoir s'orienter dans un site web en ayant parfaitement connaissance de l'ensemble des éventuelles alternatives.

Bien que conçus au départ pour les personnes non-voyantes, les plans de sites générés peuvent également aider tout internaute, handicapé ou non, notamment les personnes souffrant de déficiences cognitives ou de déficits de mémorisation. De plus, notre générateur peut apporter une aide précieuse aux webmestres désirant réorganiser leurs sites web. Ils visualiseront ainsi une représentation thématique de leur site, et pourront faciliter la navigation en définissant un nombre maximum de liens à chaque niveau de l'arborescence. Précisons, que le webmestre peut également choisir de générer le plan réel de son site ou le plan par thème (ou encore celui qui va nous intéresser par la suite, à savoir le plan par thème en définissant un nombre un maximum de liens à chaque niveau de l'arborescence). Quel que soit le plan généré, il a de plus la possibilité de choisir l'affichage sous forme de liste ou bien d'opter pour un affichage graphique (figure 5.9).

Afin de résoudre ce problème de transformation de graphe en arbre, nous avons utilisé un algorithme de fourmis artificielles inspiré de l'approche de Reimann et Laumanns [112] (section 3.4 page 92). Nous avons adapté cette méthode à notre problématique de générateur de plan de site web pour les personnes handicapées visuelles.

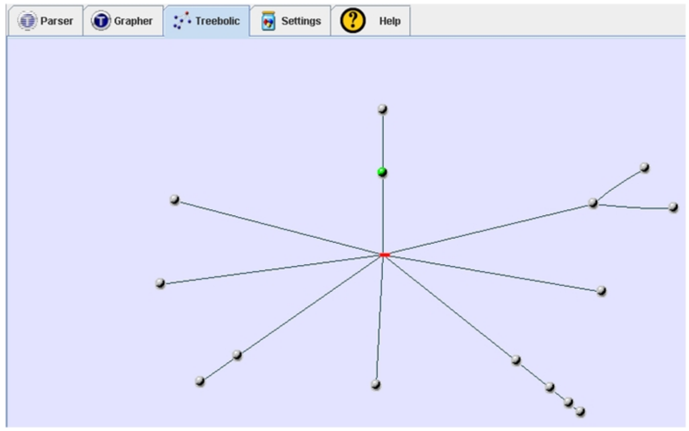


FIG. 5.9 – Affichage graphique proposé aux webmestres.

5.3.1 Modèle biologique et informatique des fourmis artificielles

Reimann et Laumanns [112] utilisent un algorithme de type ACO pour résoudre le « Capacitated Minimum Spanning Tree » (CMST) présenté dans la section 3.4. Les fourmis artificielles ont pour rôle de déterminer des regroupements optimums de nœuds du graphe. Une fois les regroupements effectués, le sous-problème restant consiste à trouver l'arbre de poids minimum, ce qui peut être réalisé par les algorithmes de Kruskal [73] ou de Prim [106]. En prenant pour base l'algorithme ACO, nous avons réalisé un générateur de plan de site web pour personnes handicapées. Il est effectivement aisé de faire l'analogie entre la structure d'un site web et un terrain composé du nid de la fourmilière et de plusieurs sources de nourriture. Pour cela, considérons un réseau de sources de nourriture connectées entre elles par des chemins, ainsi que des fourmis circulant entre ces sources. Nous remarquons que les fourmis peuvent utiliser les chemins existants ou en créer de nouveaux pour relier au mieux les différents lieux de nourriture. Chaque élément peut être confronté à son homologue biologique qu'il modélise, à savoir :

- le nid : il correspond à la page d'accueil de notre site web. Le nid est essentiellement composé de fourmis et d'une zone permettant de stocker la nourriture ainsi que des informations sur la distance parcourue pour collecter les aliments. Dans notre architecture, cela revient à mémoriser le chemin parcouru pour atteindre une page web ;
- la nourriture : chaque page web contient une certaine quantité d'informations. Cette information est toujours disponible à chaque visualisation de la page par l'internaute. Par analogie, chaque page web représente une source de nourriture renouvelable pour les fourmis ;
- les chemins : ils représentent quant à eux les liens entre les pages. Ajoutons qu'un chemin sera caractérisé par sa longueur — laquelle représente la dissimilarité entre les pages web — et qu'il sera marqué de phéromones s'il participe à l'élaboration d'une solution de notre problème. Plusieurs calculs de dissimilarité entre les pages web sont envisageables. Nous avons utilisé un TF/IDF [115]. Tout d'abord un seuil minimal est posé pour filtrer les termes de poids trop faible. Pour chaque document i composant

le site web, on calcule un vecteur $tfidf_i$, où chaque composante est notée $tfidf_{i,j}$ avec j un terme du document i. Précisons que chaque composante peut se calculer ainsi :

$$tfidf_{i,j} = \frac{tf_{i,j}}{df_j} \tag{5.1}$$

où $tf_{i,j}$ (pour « term frequency ») représente la fréquence d'apparition du terme j dans le document i et df_j (pour « document frequency ») représente la fréquence d'apparition du terme j dans l'ensemble des documents du site web. Pour calculer la dissimilarité entre deux documents du site web, une mesure de cosinus est réalisée entre les vecteurs *tfidf* représentatifs de deux documents. Ainsi, plus les pages sont proches thématiquement, plus le chemin les reliant est court.

La figure 5.10 (a) représente la structure d'un site web composé de la page d'accueil (le nid) et de 3 pages web (les 3 sources de nourriture).

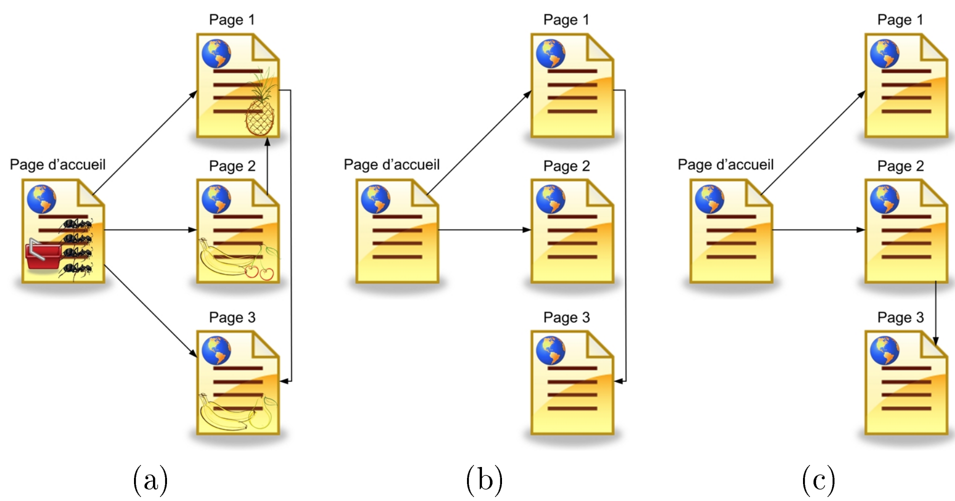


Fig. 5.10 – (a) La fourmilière et 3 sources de nourriture ; les solutions de plans de site avec au maximum deux liens sortants : utilisation de liens déjà existants (b), et création de nouveaux liens (c).

Les fourmis doivent marquer des chemins avec des phéromones pour atteindre chaque source de nourriture par une voie unique, la plus courte possible. Dès lors, un plan de site possible serait de garder trois liens partant de la page d'accueil vers les trois pages. Dans notre approche, c'est précisément l'utilisateur qui détermine le nombre de liens maximum à chaque niveau. Or s'il souhaite n'avoir au maximum que deux liens, ce n'est pas une solution réalisable. La figure 5.10 (b) et (c) illustre deux solutions possibles (parmi d'autres). Sur la figure 5.10 (b) seuls des chemins déjà existants ont été conservés (utilisation de liens existants), alors que sur la figure 5.10 (c) un chemin reliant la page 2 à la page 3 a été créé par les fourmis (présentation d'un lien dans le plan de site bien qu'il soit inexistant dans la structure réelle du site web).

5.3.2 Algorithme principal du générateur de plan

Notre méthode de générateur de plan est composée de trois modules distincts : premièrement l'analyse du site, deuxièmement la génération du plan, et troisièmement l'affichage du plan. Dans la première partie, le site web est analysé (génération d'un graphe représentant le site) et la similarité entre les pages web est évaluée (laquelle définit la distance entre les sources de nourriture). A ce stade, la génération du plan peut débuter. Le principe général de l'algorithme de génération de plan de site est donné par l'algorithme 4.

Le rôle des fourmis consiste à générer un nombre h de circuits. Les circuits représentent des regroupements de sources de nourriture proches les unes des autres. Une fois les h circuits générés, toutes les sources de nourriture doivent être accessibles. En partance de chaque source de nourriture i, h_i circuits sont générés. Ce nombre h_i de circuits est une variable dépendant des capacités de mémorisation de l'utilisateur. Au début de la génération du plan, l'internaute précise la valeur de la variable K_{liens} qui correspond au nombre maximum de liens sortant de chaque nœud de l'arbre qu'il désire obtenir dans son plan de site. Cette valeur de K_{liens} correspond au nombre maximum de chemins autorisés sortant d'une source de nourriture afin de joindre toutes les sources de nourriture. Ainsi le nombre h_i de circuits générés en partance d'une source de nourriture par chaque fourmi est tel que $h_i \leq K_{liens}$.

Algorithme 4 Génération de plan de site avec des fourmis artificielles

1: **Initialiser** les valeurs des phéromones τ_{ij}
2: **Répéter**
3:     **Pour toutes** les fourmis
4:     Générer les h circuits
5:     **Pour tout** circuit **Faire**
6:         Générer un arbre avec l'algorithme de d-Prim
7:     **Fin pour**
8:     **Fin Pour**
9:     Mettre à jour la meilleure solution trouvée
10:     Mettre à jour la matrice de phéromones τ_{ij}
11: **Jusqu'à** la validation du test d'arrêt
12: **Retourner** la meilleure solution trouvée depuis le début

L'algorithme de Prim permet de construire un arbre couvrant de poids minimum à partir d'un graphe connexe G avec une valuation positive des arêtes (section 3.4.2 page 93). Dans notre méthode, l'algorithme d-Prim (présenté dans la section 3.4.2 pour résoudre le d-MST) est utilisé sur des sous-graphes du graphe initial représentant le site web (ou plus précisément sur les regroupements de pages réalisés par les fourmis artificielles).

Dans la section suivante sera détaillé le rôle des fourmis dans cet algorithme. Dans un premier temps la génération des circuits sera expliquée (algorithme 4, ligne 4), puis la mise à jour de la matrice de phéromone sera présentée (algorithme 4, lignes 1 et 10), pour finir par la vérification du critère d'arrêt (algorithme 4, ligne 11).

5.3.3 Rôle des fourmis artificielles

Génération des h circuits

Lors de la construction d'une solution réalisable, la fourmi doit opérer des choix concernant le chemin à emprunter pour rejoindre une source de nourriture. Ceux-ci sont influencés par la distance séparant les sources de nourriture et par la quantité de phéromones déposées par ses congénères. Cette règle de décision peut donc être notée comme suit :

$$P_{ij} = \begin{cases} \frac{s_{ij}\tau_{ij}}{\sum_{(i,l)\in\Omega} s_{il}\tau_{il}} & \text{si } (i,j) \in \Omega \\ 0 & \text{sinon} \end{cases} \tag{5.2}$$

où Ω représente l'ensemble des chemins possibles permettant d'atteindre les sources de nourriture qui ne sont pas encore rattachées à un circuit. Dans l'équation 5.2, P_{ij} donne la probabilité de rassembler les sources de nourriture i et j dans un même circuit. τ_{ij} représente la quantité de phéromones présentes sur le chemin reliant la source de nourriture i à la source de nourriture j. s_{ij} représente l'économie réalisée par le regroupement des sources i et j. La figure 5.11 permet de donner un exemple pour le calcul de l'économie, où d_{ij} équivaut à la distance entre les sources i et j. Il s'agit du cas où la fourmi se trouve sur un nœud appelé O (son nœud de départ) et doit décider si elle souhaite regrouper i et j dans un même circuit ou non.

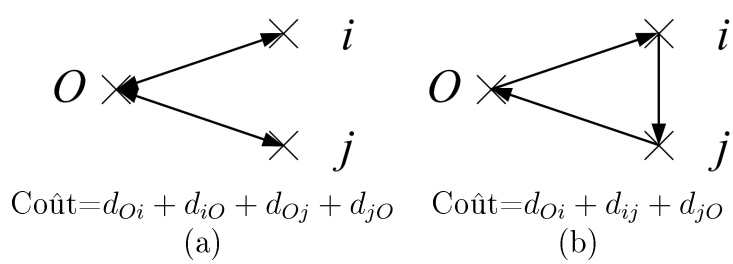


FIG. 5.11 – Exemple de calcul de l'économie s_{ij}.

Initialement, le nœud O correspond au nid et chaque source de nourriture est accessible depuis le nœud O. Le coût de cette solution correspond à la somme des distances pour relier O à chaque source et pour retourner au nœud O (figure 5.11 (a)). La figure 5.11 (b) représente la configuration où la source j est placée à la suite du nœud i dans un même circuit. Le coût de cette solution va donc devenir la somme des distances pour parcourir le circuit (Oij). Finalement l'économie s_{ij} réalisée en faisant un lien de i vers j dans un même circuit, correspond à la différence des deux coûts présentés dans la figure 5.11, et est donnée pour cet exemple par la formule 5.3.

$$s_{ij} = d_{iO} + d_{Oj} - d_{ij} \tag{5.3}$$

Lorsque la fourmi construit son trajet, elle devra également veiller à ne pas trop se disperser — le nombre de chemins directement en partance d'un nœud i étant limité par la valeur de K_{liens}. Ainsi lorsque la fourmi décide de se rendre à une nouvelle source de nourriture, et, connaissant a priori les autres destinations possibles, elle peut choisir de chercher sa destination :

- à partir de l'extrémité d'un chemin déjà commencé,
- à partir d'un nœud de départ i si le nombre h_i de chemins déjà générés à partir du nœud i est inférieur à K_{liens}. En effet, si $h_i = K_{liens}$, alors aucun autre circuit ne peut être généré à partir du nœud i, donc $\forall j \; (i, j) \notin \Omega$.

Construisant ainsi Ω, il ne lui reste plus qu'à faire son choix, lequel sera influencé — comme le montre l'équation 5.2 — par l'économie qui dépend de la longueur des chemins, et par la quantité de phéromones sur les chemins.

La figure 5.12 permet d'illustrer la génération des circuits par une fourmi. Supposons que l'utilisateur ait fixé $K_{liens} = 3$. Dans cet exemple, on trouve 8 sources de nourriture et le nid. Imaginons le travail d'une fourmi initialement placé sur le nid O.

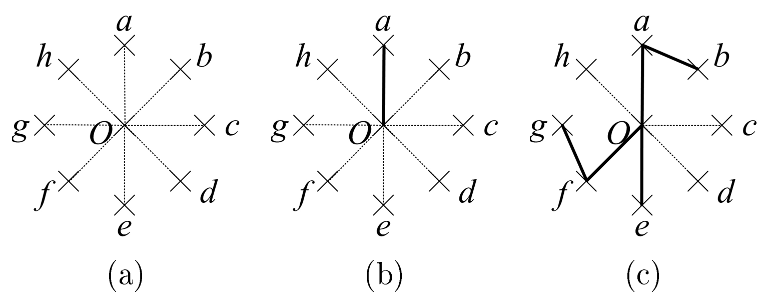


FIG. 5.12 – Génération de circuits par les fourmis artificielles : (a) initialement, tous les nœuds sont reliés au nid, (b) la fourmi a décidé de relier 'a' au nid, (c) il reste 3 nœuds ('c', 'd' et 'h') à relier à la suite de circuits déjà existants.

Au départ, pour calculer le coût de cette solution, on considère que chaque source est liée au nid O (figure 5.12 (a)) et chacune est capable de débuter un circuit. On a alors $\Omega = \{(Oa), (Ob), (Oc), (Od), (Oe), (Of), (Og), (Oh)\}$. La fourmi doit ensuite choisir quel sera le début de son premier circuit. Pour cela on calcule $P_{ij} \; \forall \; (i, j) \in \Omega$. La figure 5.12 (b) illustre le cas où la fourmi aurait choisi de se diriger vers la source de nourriture a. A cet instant, $h_O = 1 \leq K_{liens}$. La fourmi peut donc continuer son circuit commençant par a ou créer un nouveau circuit à partir de O, ce qui se traduit par :
$\Omega = \{(Ob), (Oc), (Od), (Oe), (Of), (Og), (Oh), (ab), (ac), (ad), (ae), (af), (ag), (ah)\}$.
Elle est alors influencée par la quantité de phéromones sur les chemins ainsi que par l'économie potentiellement réalisée. La figure 5.12 (c) représente cette même génération de circuits après plusieurs choix de la fourmi. Ici, trois circuits sont en construction : (Oab), (Oe), et (Ofg). On a donc $h_O = 3$. La limite du nombre de circuits à générer à partir de O est atteinte. Il reste trois sources de nourriture à placer, mais il n'est plus possible de générer de nouveau circuit à partir du nid O. On a donc $\Omega = \{(ac), (ad), (ah), (bc), (bd), (bh), (ec), (ed), (eh), (fc),$

$(fd), (fh), (gc), (gd), (gh)\}$. La fourmi devra compléter ainsi les circuits déjà commencés ou en créer à partir de nœuds appartenant déjà à un circuit, jusqu'à ce que $\Omega = \emptyset$.

Le rôle d'une fourmi est de rapporter au nid de la nourriture provenant de toutes les sources. Elle ne doit accéder à une source que par un unique chemin, le plus court possible. Une fois sa tâche accomplie, l'algorithme d-Prim est exécuté sur chacun des circuits afin de générer des sous-arbres de poids minimum qui valident la contrainte du nombre maximum de liens à chaque niveau de l'arborescence inférieur à K_{liens}. Une solution réalisable de notre problème de génération d'arbre respectant la contrainte K_{liens} est alors obtenue. Une fois que toutes les fourmis ont construit leur solution, seule la meilleure sera mémorisée (algorithme 4, ligne 9).

Les phéromones

A chaque fois que toutes les solutions réalisables ont été générées, la quantité de phéromones sur les chemins est mise à jour. Celle-ci influence le choix des fourmis sur le chemin à utiliser pour se rendre à une source de nourriture. Le choix d'un chemin dépend effectivement de la distance minimale possible pour atteindre ladite source mais également de la quantité de phéromones présentes sur le trajet. Au premier passage de la colonie, les fourmis seront influencées par les phéromones présentes sur les chemins initialement connus. Aux itérations suivantes, elles seront influencées par les phéromones déposées sur les chemins constituant la meilleure solution jusqu'alors déterminée.

La mise à jour des phéromones est donnée par la formule 5.4 :

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau^*_{ij} \tag{5.4}$$

Avec ρ le coefficient d'évaporation et $\Delta\tau^*_{ij}$ le coefficient de renforcement des phéromones, calculé pour favoriser le bon choix des fourmis. Dans la littérature on trouve généralement pour le calcul de $\Delta\tau^*_{ij}$ la formule donnée par l'équation 5.5 :

$$\Delta\tau^*_{ij} = \begin{cases} \frac{1}{c(S^*)} & \text{si } (i,j) \in S^* \\ 0 & \text{sinon} \end{cases} \tag{5.5}$$

où S^* représente la meilleure solution trouvée et $c(S^*)$ le coût de cette solution. Ainsi pour les liens (i,j) n'appartenant pas à la meilleur solution S^*, nous aurons $\Delta\tau^*_{ij} = 0$. La quantité de phéromones τ_{ij} sur ces chemins s'évapore donc jusqu'à tendre vers 0. Les liens (i,j) appartenant à la meilleure solution S^* sont tels que $\Delta\tau^*_{ij} > 0$, ce qui permet de renforcer la quantité de phéromones. Et cette quantité de phéromones augmente d'autant plus que le coût $c(S^*)$ est faible. Afin d'obtenir une quantité de phéromones τ_{ij} qui ne dépende pas de la fonction coût, et donc qui soit indépendante du site web, nous avons utilisé la fonction $\Delta\tau^*_{ij}$ donnée par l'équation 5.6 :

$$\Delta\tau^*_{ij} = \begin{cases} 1 & \text{si } (i,j) \in S^* \\ 0 & \text{sinon} \end{cases} \tag{5.6}$$

Par conséquent, les chemins qui appartiennent à la meilleure solution ont des arcs dont la quantité de phéromones tend vers 1 tandis que ceux qui ne participent pas à la meilleure solution ont des arcs dont la quantité de phéromones tend vers 0. A l'initialisation, avec pour finalité de favoriser les liens existants dans le site web, nous avons affecté à la matrice τ la matrice d'adjacence du site web. Cependant les fourmis doivent être capables de générer d'autres chemins s'ils s'avèrent plus avantageux en termes d'économie s_{ij}. De ce fait nous avons fixé τ_{ij} à une faible valeur non nulle pour les liens (i, j) ne faisant pas partie du graphe du site web.

Le test d'arrêt

La meilleure solution S^* trouvée par les fourmis constitue une solution réalisable de notre problème. La quantité de phéromones augmente pour les liens appartenant à S^* et diminue dans le cas contraire. De plus, au fur et à mesure des itérations — les fourmis étant attirées vers les chemins contenant la plus grande quantité de phéromones —, un écart se creuse au niveau de la quantité de phéromones sur les liens, selon qu'ils appartiennent à la solution ou non. Lorsque cet écart devient significatif, c'est-à-dire lorsque :

1. $\tau_{ij} < \alpha \ \forall (ij) \notin S^*$
2. et $\tau_{ij} > \beta \ \forall (ij) \in S^*$,

le test d'arrêt est vérifié, et la meilleure solution peut être déterminée, en vue de son affichage sous forme linéaire.

5.3.4 Application et résultats expérimentaux

L'algorithme proposé a été programmé en Java et exécuté sur un Pentium à 1.6 GHz. Pour l'algorithme des fourmis artificielles, nous avons utilisé les paramètres suivants : une population de n fourmis, avec n correspondant au nombre de pages du site web ; $\rho = 0.1$; $\alpha = 0.5$; $\beta = 0.8$.

La figure 5.13 représente la quantité de phéromones sur les liens reliant la page d'accueil O du site à toutes les autres pages — numérotées à partir de 1 — lors de la génération d'un plan de site avec $K = 4$. Précisons que seuls les liens présentant des évolutions particulières — c'est-à-dire faisant partie à un moment donné de la meilleure solution — sont représentés sur le graphique. Nous noterons E les liens pour lesquels les phéromones ne cesseront de s'évaporer. Nous distinguons sur la figure les liens E_1 qui possédaient initialement une forte quantité de phéromones, à l'opposé des liens E_2 qui ne disposaient au départ que d'une faible quantité.

Lors de cette exécution, les liens $(O3)$, $(O4)$ et $(O5)$ ont été sélectionnés par les fourmis lors des quatre premières itérations. La quantité de phéromones sur ces liens étant initialement fixée à 1, elle reste constante sur les quatre premières itérations. Lors des deux itérations suivantes, les liens $(O1)$, $(O5)$, $(O9)$ et $(O15)$ appartiennent à la meilleure solution jusqu'alors trouvée par les fourmis. Sur ces liens, de nouvelles phéromones sont déposées alors que sur tous les autres les phéromones s'évaporent durant ces deux itérations. Lors des

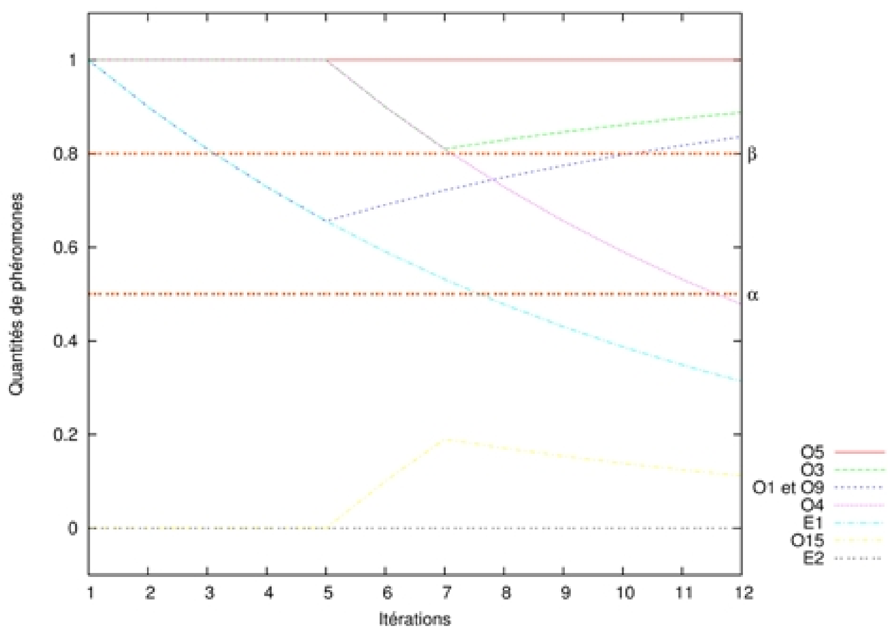


FIG. 5.13 – Quantité de phéromones sur les liens partant de la racine lors de la génération d'un plan du site de Polytech'Tours avec $K = 4$.

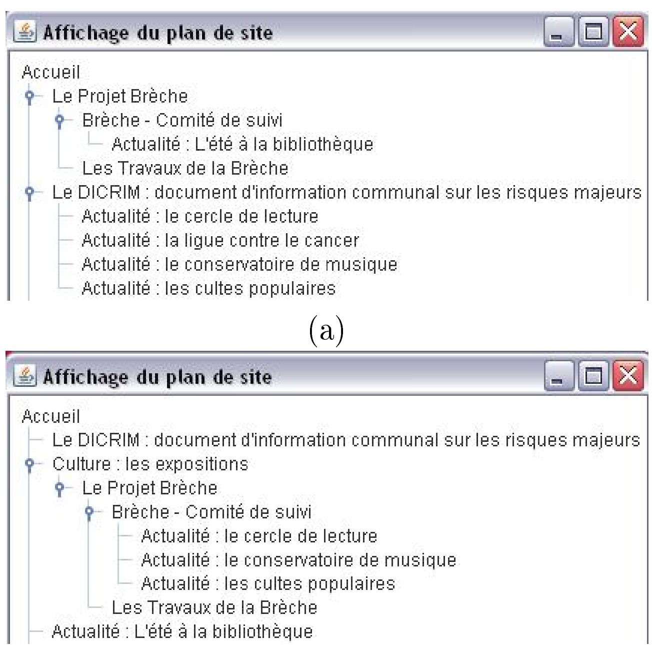
itérations suivantes ce sont les liens ($O1$), ($O3$), ($O5$) et ($O9$) qui font partie de la meilleure solution. La quantité de phéromones sur ces quatre liens va donc augmenter tandis qu'elle va diminuer sur tous les autres liens. A l'itération 12, les liens faisant partie de la meilleure solution sont toujours les liens ($O1$), ($O3$), ($O5$) et ($O9$). Après mise à jour des phéromones, nous observons que ces quatre liens possèdent une quantité de phéromones supérieure à β. De plus, tous les autres liens possèdent une quantité de phéromones inférieure à α. Le test d'arrêt est alors vérifié pour les liens de premier niveau.

Nous allons désormais présenter les résultats obtenus sur différents sites web : ville de Langeais (15 pages), ville de Chinon (26), Polytech'Tours (126 pages), et l'équipe HaNT du Laboratoire d'Informatique de l'Université de Tours (149 pages). Le tableau 5.1 représente les résultats obtenus pour l'exécution du programme de génération de plan de site web par des fourmis artificielles, en comparaison avec une méthode aléatoire et l'algorithme d-Prim. Le coût des solutions est calculé comme la somme des similarités entre les pages liées entre elles dans le plan du site. Le coût moyen suivi de l'écart moyen entre parenthèses est indiqué et la durée d'exécution est donnée en ms. Ce tableau permet de comparer les résultats obtenus sur plusieurs sites web pour lesquels le nombre n de pages est indiqué. Dans cette série de tests $K = 4$, c'est-à-dire qu'à tous les niveaux du plan du site, 4 liens maximum sont attendus par l'utilisateur. Nous observons que notre algorithme de fourmis artificielles donne des résultats avoisinant les résultats obtenus par l'algorithme d-Prim. Cependant nous remarquons également que la durée d'exécution est bien plus importante que pour l'algorithme d-Prim et cela se ressent de plus en plus lorsque le site web contient de nombreuses pages.

	Fourmis artificielles		d-Prim		Aléatoire	
n	Coût	Durée	Coût	Durée	Coût	Durée
15	4.52 (0.15)	30	3.98	15	5.04 (0.31)	15
26	11.22 (0.25)	39	8.59	16	12.51 (0.37)	15
126	118.90 (0.15)	75158	118.35	18	144.08 (1.36)	17
149	60.05 (3.63)	148501	47.05	18	124.72 (0.27)	17

TAB. 5.1 – Coûts (et écarts moyens) sur différents sites web avec K=4.

La figure 5.14 (a) illustre le résultat obtenu grâce à l'algorithme d-Prim sur le site web de la ville de Chinon avec $K = 4$. La figure 5.14 (b) illustre quant à elle un des résultats obtenus sur le même site web avec l'algorithme de fourmis artificielles. Nous retrouvons sur ces deux plans des liens concernant l'actualité qui sont regroupés, de même que les liens concernant le projet Brèche. Cependant, dans les deux cas, les liens concernant l'actualité ont une position qui semble étrange dans l'arborescence du plan du site. En effet ces algorithmes tentent de minimiser le coût de l'arbre, utilisant pour cela une mesure de dissimilarité entre les pages web. La mesure utilisée ici est une mesure TF/IDF qui ne donne pas des résultats satisfaisants pour une application concrète à notre problème de plan de site web.



(a)

(b)

FIG. 5.14 – Affichage du plan du site de la ville de Chinon généré avec l'algorithme d-Prim (a) et avec des fourmis artificielles (b) pour $K = 4$.

5.4 Conclusion

Finalement, ce chapitre nous a permis de présenter deux outils de synthèse.

Dans un premier temps, nous avons réalisé un validateur des normes d'accessibilité destiné aux webmestres débutants. D'une part, il s'adapte parfaitement à une démarche progressive de correction grâce au regroupement des erreurs d'une ou plusieurs pages du site. D'autre part, la particularité de notre validateur réside dans sa capacité à tester de nombreuses recommandations, qu'elles soient HTML, CSS ou encore des recommandations d'accessibilité telles les normes WCAG, Accessiweb ou UWEM. Les fonctionnalités multi-pages et multi-recommandations permettent ainsi aux webmestres de n'utiliser qu'un seul validateur dans toute leur démarche. Ce validateur est utilisable à l'adresse `http://accessi.hant.li.univ-tours.fr/validateur/accueil.php`. Notons qu'avec l'arrivée de nouveaux pays européens adhérant aux normes UWEM, des recommandations nationales autre que Accessiweb pourront être ajoutées. A ce jour, il est d'ors-et-déjà possible d'intégrer les recommandations espagnole et belge. A cela, devront aussi être ajoutées les normes WCAG 2.0 (successeur de WCAG 1.0).

Dans un second temps, nous avons présenté un algorithme de fourmis artificielles pour la génération de plans de sites web. L'intérêt majeur de cet algorithme consiste à pouvoir s'adapter aux capacités de mémorisation des internautes non-voyants en leur octroyant la liberté de paramétrer le nombre K de liens maximum à chaque nœud de la structure du plan. A partir de ce travail, des améliorations sont envisageables d'une part pour améliorer les performances et la vitesse d'exécution et d'autre part car la génération du plan du site par cet algorithme est directement liée à l'analyse de la similarité entre les pages. Dans ce travail, la mesure TF/IDF a été utilisée et les résultats sont donc comparés non pas aux pages du site mais à la matrice de similarité entre les pages. Par conséquent, d'autres mesures (permettant de détecter un thème commun même si les termes utilisés diffèrent) pourraient être mises en place afin d'améliorer la qualité du plan dans des conditions réelles d'utilisation.

Les travaux présentés dans ce chapitre ont donné lieu à l'encadrement de deux projets de fin d'études au Département d'Informatique (DI) de Polytech'Tours [90][130] et d'un stage de Master 2 Recherche [129]. Ils ont également conduit à un certain nombre de communications ou de publications [33][38][43][42].

Chapitre 6

Outils de transformation de pages

Ce chapitre présente deux outils de transformation de pages destinés à faciliter l'accès à l'information pour les internautes handicapés. Le premier consiste à obtenir une page accessible à partir d'une page non-accessible, et le deuxième consiste à réorganiser les différents éléments qui composent une page.

6.1 Introduction

Ce chapitre concerne les outils de transformation des pages web.

Dans le domaine de l'accessibilité, la première transformation de la page à réaliser consiste à modifier le code source de la page afin qu'il respecte les normes d'accessibilité. Nous avons à ce sujet réalisé un correcteur de pages web pour assister les webmestres dans le processus de normalisation qu'ils ont entamé. Précisons que ce correcteur peut également être utilisé par les internautes pour transformer automatiquement les pages qu'ils visitent en pages « plus accessibles ».

Bien qu'accessibles, les sites web peuvent être difficilement lisibles par les personnes utilisant un lecteur d'écran [131]. Il est clair qu'avec un tel dispositif, la lecture est linéaire et cela implique notamment la lecture — ou la relecture pour les pages suivantes — du titre, de la bannière publicitaire etc. avant celle du contenu. L'internaute, qu'il soit handicapé ou non, doit pouvoir distinguer l'essentiel de l'accessoire (principe de prégnance) et structurer les informations de telle sorte que celles disponibles soient hiérarchisées. Nous avons alors réalisé un réorganisateur qui permet de distinguer les différentes zones contenues dans une page web (tel que le fait un internaute visuellement) et de choisir celles qu'il souhaite afficher ainsi que leur ordre.

6.2 Correcteur de pages web

Les webmestres novices en matière d'accessibilité sont à même de rencontrer des difficultés pour rendre leur site web accessible. Ils peuvent s'informer en lisant par exemple les normes officielles ou les guides d'aide à l'accessibilité disponibles sur Internet. Ils peuvent également repérer leurs erreurs grâce à des validateurs automatiques. Si certaines sont facilement corrigibles, d'autres demandent en revanche une plus grande quantité de travail. Nous avons de ce fait réalisé un correcteur de page web. Notons que pour faciliter l'évolutivité de cet outil, nous l'avons créé sous la forme d'une plateforme de traitements sur des pages web. Cela permet de créer et d'utiliser des traitements facilement sans avoir à modifier entièrement le programme. Nous avons donc séparé notre application en deux parties distinctes : premièrement, le moteur de transformation de pages web, et deuxièmement des plugins gérant chacun un traitement. Grâce à cette plateforme, l'utilisateur peut lister les traitements disponibles et choisir ceux qu'il souhaite appliquer sur sa page web. La figure 6.1 présente un synoptique de ce que pourra réaliser l'utilisateur.

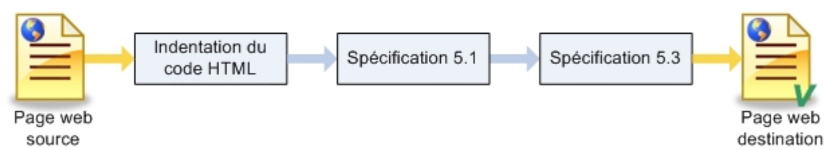


FIG. 6.1 – Exemple de succession de traitements.

Dans cet exemple, l'utilisateur applique trois traitements successifs sur sa page web. Le premier permet d'indenter le code de la page, puis le deuxième et le troisième appliquent respectivement les corrections correspondant aux spécifications 5.1 et 5.3 des WCAG.

6.2.1 Le moteur

Le but du moteur est simple : il consiste à charger une page web, à la modifier puis à la restituer. Ce moteur est disponible en deux versions :

1. une version console qui peut être utilisée par un autre programme, un serveur ou un proxy (la figure 6.2 représente ce dernier cas ou l'internaute reçoit une page valide après correction automatique),
2. une version avec une interface (figure 6.3) qui peut être utilisée par le webmestre pour transformer sa page web.

La page web source (par URL ou par chargement de fichier) doit être indiquée au moteur. La version avec interface possède un champ spécifique à renseigner tandis que dans le mode console, cette information est soumise au moteur en ligne de commande. Dans la version avec interface, deux types de sorties sont disponibles. D'une part, en cliquant sur le bouton « Générer Aperçu » la page web source et/ou la page web de sortie peuvent être visualisée.

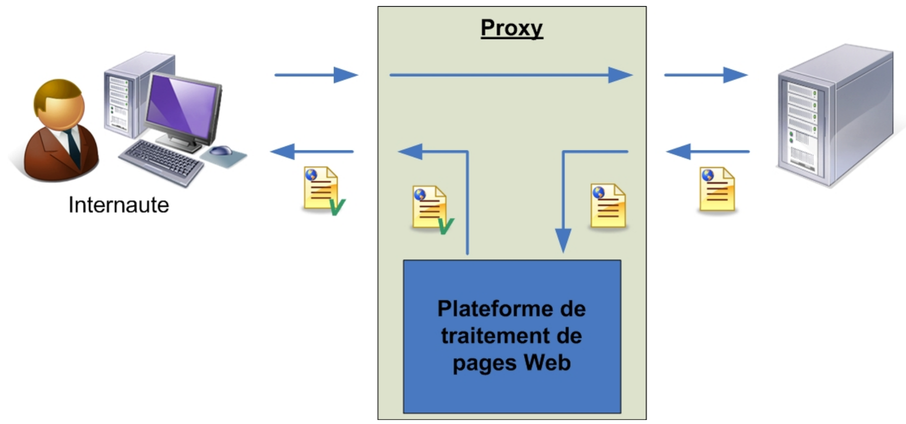


FIG. 6.2 – Fonctionnement en mode console avec un proxy.

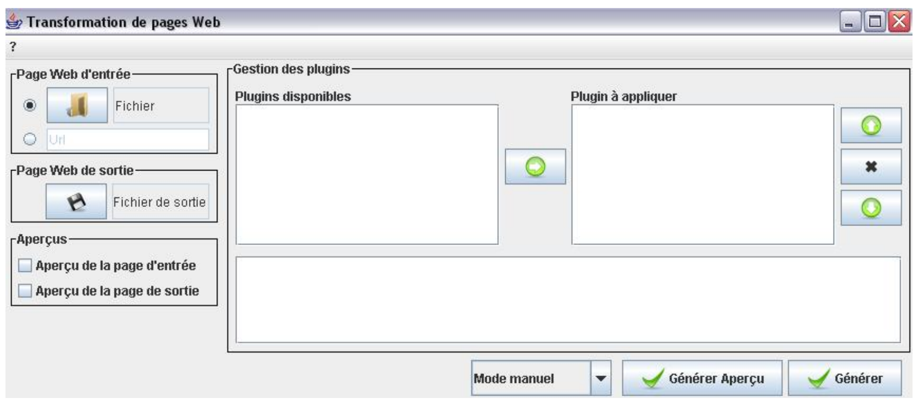


FIG. 6.3 – Interface du moteur.

Dans ce cas les modifications effectuées par les différents plugins sur la page source ne seront pas sauvegardées. D'autre part, pour garder une trace des différentes modifications effectuées, la page de sortie peut être enregistrée en local au format html. En mode console, seule ce deuxième type de sortie est disponible en indiquant le fichier de sortie au moteur en ligne de commande.

Précisons que le moteur permet d'effectuer les traitements soit en mode automatique soit en mode manuel. Lors de l'utilisation de la version avec interface ces deux modes de transformation sont possible, tandis qu'avec la version en mode console sur un proxy n'effectuera les transformations qu'en mode automatique (donc de manière transparente pour l'internaute). Cependant cette fonctionnalité dépend des traitements. Certains correspondent en effet à des spécifications WCAG qui ne peuvent être effectuées automatiquement et nécessitent une intervention humaine. Dans ce cas ils ne pourront pas être utilisés lors de la correction automatique *via* un proxy, ni lors de la correction automatique en mode interface.

6.2.2 Les plugins

Les plugins correspondent à divers traitements qu'il est possible d'accomplir sur une page web. Il peut s'agir d'effectuer des statistiques sur la page, d'indenter le code source... L'enjeu ici est de réaliser des plugins afin de rendre les pages web accessibles. Nous avons de ce fait créé un plugin par spécification des WCAG, chacun ayant ainsi pour rôle de corriger les pages afin qu'elles respectent la spécification qu'il représente. Dans ce but, nous avons étudié les spécifications WCAG afin de déterminer :

- si une correction ne requiert que l'intervention humaine : dans ce cas il n'est pas utile de créer un plugin ;
- si la correction peut se faire de manière semi-automatique : dans ce cas un plugin peut être créé pour faciliter la correction. Ce type de plugin doit s'exécuter par le moteur en mode « manuel » ;
- si la correction peut se faire de manière automatique : dans ce cas un plugin doit être créé. Il pourra s'exécuter en mode « automatique » pour réaliser la transformation sans aucune intervention humaine, ou en mode « manuel » pour laisser le webmestre choisir parmi plusieurs configurations possibles de transformation.

D'un côté, lors du lancement du moteur en mode interface, celui-ci recherche dans un dossier spécifique les plugins disponibles pour les proposer à l'utilisateur, lequel doit alors choisir dans cette liste, ceux qu'il désire exécuter ainsi que l'ordre d'exécution. La figure 6.4 présente l'interface du moteur avec des plugins à disposition. Dans cet exemple l'utilisateur a choisi d'appliquer le plugin correspondant à la spécification 5.1 des WCAG.

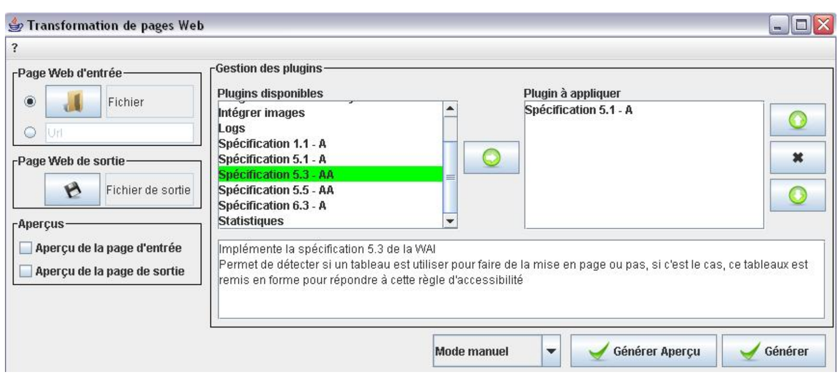


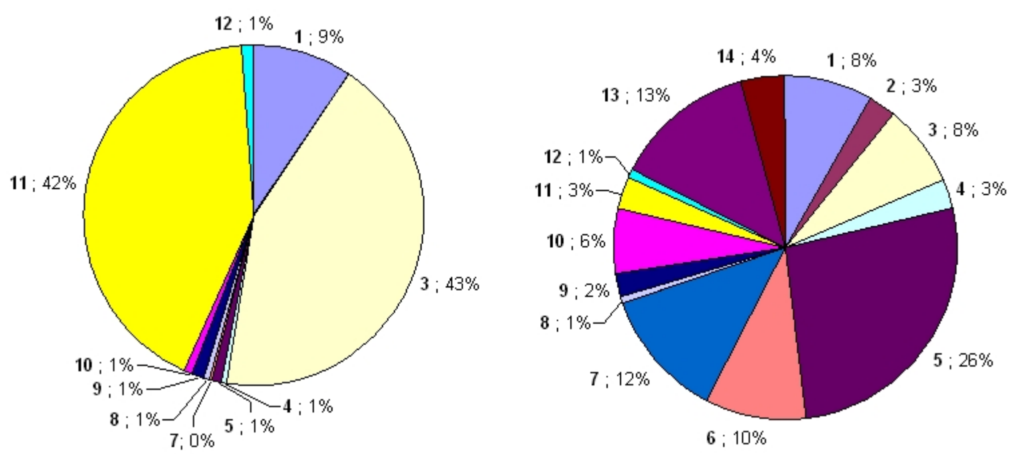
FIG. 6.4 – Interface du moteur ayant chargé les plugins.

D'un autre côté, lors de l'exécution en mode console, la liste des plugins à exécuter est stockée dans un fichier XML que le moteur lit avant de commencer les traitements. Ce fichier peut comporter la liste complète des plugins disponibles ou ne contenir que ceux correspondant à des spécifications qui présentent des erreurs dans la page. Ceci peut être le cas si le correcteur est exécuté à la suite d'un validateur qui aura préalablement repéré les spécifications à valider.

6.2.3 Choix des plugins implémentés

Pour que notre correcteur soit complet, l'idéal serait d'implémenter un plugin par spécification WCAG. Comme nous l'avons précédemment précisé, nous avons choisi de répertorier les spécifications en trois catégories afin de déterminer si le plugin associé peut être exécuté en mode manuel et automatique, ou uniquement en mode automatique, ou bien s'il n'est pas possible d'automatiser la correction. Ce dernier cas représente en réalité les spécifications pour lesquelles trop de renseignements seraient demandés au webmestre, ce qui rendrait l'application trop contraignante à l'utilisation. Mise à part ces dernières spécifications, toutes peuvent trouver leur place dans notre correcteur.

Pour commencer, nous avons choisi de réaliser les plugins représentant les spécifications engendrant le plus grand nombre d'erreurs. Notre application étant initialement destinée aux webmestres des collectivités locales — afin de compléter l'outil d'aide à l'accessibilité (section 4.2) et le validateur multi-recommandations (section 5.2) —, nous avons évalué l'accessibilité par rapport aux normes WCAG d'un échantillon de 50 sites web de communes d'Indre-et-Loire. Notons que les résultats de cette analyse sont donnés dans l'annexe B. A savoir, d'une part, on peut y voir que 4631 erreurs ont été détectées se répartissant essentiellement (à 94%) sur les directives 3, 11, et 1 (représentant respectivement 43%,42% et 9% des erreurs). D'autre part, nous remarquons que 2604 warnings ont été relevés. Si la répartition entre les directives est plus équilibrée que pour les erreurs, la directive 5 montre tout de même le plus fort pourcentage de warnings. Les graphiques de la figure 6.5 illustrent la répartition des erreurs et des warnings entre les 14 directives. La légende de chaque portion est représentée de la manière suivante : **numéro de la directive** ; % d'erreurs ou de warnings.



(a) (b)

FIG. 6.5 – Répartition des erreurs (a) et des warnings (b) des sites web publics d'Indre-et-Loire entre les directives WCAG.

La directive 1 concerne les alternatives textuelles au contenu visuel et auditif. La spécification 1.1 étant la plus rencontrée sur les sites web, nous avons décidé de réaliser un

plugin pour celle-ci, plus précisément concernant le texte alternatif des images. La directive 3 concerne le balisage et les feuilles de style. Elle préconise notamment d'utiliser des unités relatives plutôt qu'absolues, d'user des feuilles de styles, de se servir du balisage approprié pour les listes, de ne pas utiliser de tableaux de mise en page... La directive 5 s'intéresse aux tableaux en précisant qu'il faut utiliser le balisage approprié pour les tableaux de données et que les tableaux de mise en page sont à éviter. La directive 11 a essentiellement pour rôle de définir les balises ou attributs qui sont obsolètes et qui, de ce fait, ne doivent plus être utilisés. Lister l'ensemble des attributs désuets pour les supprimer de la page web n'est pas une solution acceptable : il faudrait en effet trouver pour chaque cas une solution de remplacement.

Nous allons par la suite commencer par nous intéresser aux tableaux qui, lorsqu'ils sont utilisés en tableaux de mise en page, demandent un travail de correction conséquent au webmestre.

6.2.4 Exemples des plugins de la directive 5 des WCAG

Etude de la directive 5

La directive 5 des WCAG concerne les tableaux et s'intitule : « créer des tableaux qui se transforment de façon élégante ». Elle préconise d'utiliser les tableaux uniquement pour représenter des données existantes initialement sous forme de tableau — que nous appellerons les « tableaux de données ». Elle stipule par conséquent que les développeurs de contenus ne doivent pas utiliser les tableaux pour faire de la mise en page — nous nommerons ces types de tableaux des « tableaux de mise en page » —, ces derniers posant en effet des problèmes de lecture aux utilisateurs handicapés utilisant des dispositifs de lecture d'écran.

Cette directive comporte 6 spécifications :
- spécification 5.1 : les cellules des tableaux de données doivent être clairement identifiées. Les cellules de données pures doivent être signalées par la balise `<TD>` alors que les cellules d'en-tête du tableau doivent être marquées par la balise `<TH>`.
- spécification 5.2 : il faut utiliser les balises `<THEAD>`, `<TBODY>` et `<TFOOT>` pour regrouper les cellules de même niveau logique dans les tableaux de données comportant plusieurs niveaux logiques d'en-tête.
- spécification 5.3 : les tableaux de mise en page ne doivent pas être utilisés. Ils sont toutefois tolérés si leur lecture linéaire (de haut à gauche, en bas à droite) a un sens.
- spécification 5.4 : les tableaux de mise en page (ayant un sens en lecture linéaire, d'après la spécification 5.3), ne doivent pas comporter de balises permettant de structurer les tableaux de données pour faire de la mise en page.
- spécification 5.5 : les tableaux de données doivent posséder une courte phrase explicative en utilisant l'attribut `summary` de la balise `<TABLE>`.
- spécification 5.6 : lorsque des abréviations sont utilisées dans les en-têtes de tableaux de données, leurs significations doivent être précisées à l'aide de l'attribut `abbr` de la balise `<TH>`.

Correction des erreurs pour chacune des spécifications

Voici comment corriger les erreurs pour chacune de ces spécifications :

- spécification 5.1.
  Pour le webmestre, repérer les cellules d'en-tête dans un tableau semble trivial. Lors d'une correction automatique la distinction du rôle des cellules est plus complexe : bien souvent la première ligne du tableau correspond à son en-tête mais ce n'est pas une nécessité — il est possible en effet de trouver des tableaux avec plusieurs lignes d'en-tête ou bien à double entrée.
- spécification 5.2.
  Comme pour la spécification 5.1, il est préférable de demander au webmestre d'identifier lui-même les différents niveaux qui apparaissent dans les tableaux de données.
- spécification 5.3.
  Pour corriger cette spécification automatiquement il faut dans un premier temps être capable de distinguer les tableaux de données de ceux de mise en page ; puis, pour ces derniers, il faut restructurer les différentes cellules afin de supprimer le tableau.
- spécification 5.4.
  Comme pour la spécification 5.3, il est là aussi nécessaire de distinguer les deux types de tableaux.
- spécification 5.5.
  Il est possible de rajouter automatiquement l'attribut `summary` dans le code source de la page, mais le contenu de cet attribut devra être renseigné par le webmestre.
- spécification 5.6.
  Pour pouvoir corriger les erreurs correspondant à cette spécification, il faut être capable de détecter si un mot est une abréviation ou non, pour ensuite demander au webmestre de renseigner la signification de son abréviation.

Suite à la précédente étude des spécifications de la directive 5, nous avons effectué un classement afin de déterminer pour lesquelles une correction automatique peut être implémentée. Le tableau 6.1 représente les différentes spécifications étudiées, classées en trois catégories :

- celles qui ne peuvent pas être corrigées automatiquement,
- celles qui, au contraire, peuvent être corrigées automatiquement avec intervention du webmestre,
- enfin celles pour lesquelles une correction entièrement automatique est possible.

Correction non automatique	Correction semi-automatique	Correction automatique
5.6	5.1	5.3
	5.2	5.4
	5.5	

Tab. 6.1 – Classement des spécifications selon le type de correction qu'il est possible de réaliser.

Traitement additionnel spécifique à la directive

Afin de corriger les erreurs concernant la directive 5, une grande partie du travail consiste à être capable de détecter les tableaux de données des tableaux de mise en page. Lors de la correction en mode manuel, tous les tableaux sont montrés au webmestre et il doit lui-même distinguer les tableaux de données des tableaux de mise en page. Par contre pour le mode automatique, nous avons dû créer une règle qui permet de détecter un tableau de données. Ainsi tout tableau qui ne validera pas cette règle sera classé en tant que tableau de mise en page. Précisons que cette règle consiste à détecter si le tableau contient du code HTML dans ses cases. En effet la WAI précise qu'un tableau de données ne doit contenir que des données, c'est-à-dire pas d'images, pas de vidéos... Cette règle consiste alors à rejeter (donc à classer en tableau de mise en page) tous les tableaux contenant du code HTML dans leurs cases. Nous avons testé notre méthode de distinction[1] entre les tableaux de données et les tableaux de mise en page sur la page d'accueil des 73 sites web de l'étude e-mediacité [55] sur l'accessibilité des sites officiels. En visualisant ces pages, nous avons noté qu'aucune ne comportait de tableau de données. Notre outil a permis de dénombrer 106 tableaux principaux (les sous-tableaux ne sont pas pris en compte ici) répartis sur 38 des 73 sites. D'après nos observations, ces tableaux n'étant pas des tableaux de données, nous remarquons qu'encore 52% des sites web utilisent des tableaux de mise en page. Notre méthode de détection a permis d'en dénombrer 99 comme étant des tableaux de mise en page et 7 comme étant des tableaux de données (soit 6% d'erreurs lors de la distinction). Ces erreurs de classement étant dues dans 3% des cas à des tableaux vides et dans 3% des cas à des tableaux ne possédant pas de balises fermantes.

Interfaces de plugins de la directive 5

Nous présentons ici deux plugins : ceux des spécifications 5.3 et 5.1.

La figure 6.6 présente l'interface de correction qui s'affiche lors de l'exécution du plugin correspondant à la spécification 5.3 sur la page d'accueil du site web de l'équipe HaNT du LI de Tours. Rappelons que ce plugin permet de détecter les tableaux de mise en page et de restructurer l'information qu'ils contiennent. Cette fenêtre apparaît dès qu'un tableau de mise en page a été détecté. Un aperçu du tableau en question est visible en haut de cette fenêtre et il est possible d'insérer un titre au tableau. Dans ce dernier, la lecture linéaire est possible donc ce tableau peut être conservé sans nuire à l'accessibilité de la page. Le webmestre peut ainsi cliquer sur l'option « ne pas modifier » afin de conserver le tableau (si le tableau correspond à un tableau de données il doit agir de la même manière pour ne pas le modifier). Mais il peut également choisir de le supprimer en sélectionnant une des trois autres configurations :

1. linéarisation du tableau qui possédait un sens de lecture horizontale
2. linéarisation du tableau qui possédait un sens de lecture verticale
3. suppression du tableau en conservant le positionnement de ses différentes cases.

[1]étude réalisée en mai 2007

D'après le tableau 6.1, cette spécification peut être corrigée automatiquement c'est-à-dire sans intervention humaine. En mode automatique, c'est la 3ème configuration (conservation du positionnement des cases du tableau) qui est appliquée sur tous les tableaux de mise en page.

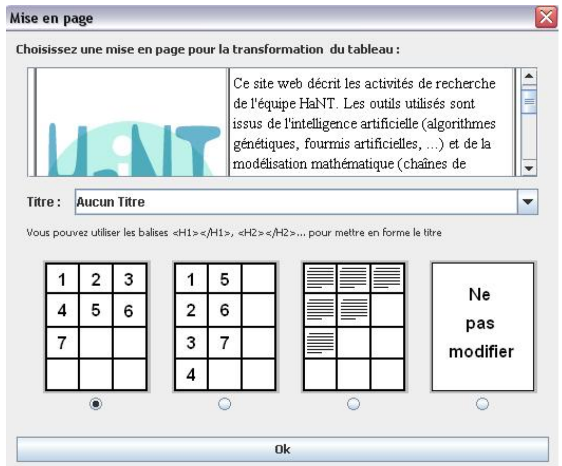


FIG. 6.6 – Interface du plugin de correction de la spécification 5.3.

Le deuxième plugin présenté est celui de la spécification 5.1 qui permet d'identifier les en-têtes des tableaux de données. Cette spécification ne peut être corrigée automatiquement et requiert toujours une intervention humaine grâce à l'interface illustrée par la figure 6.7. Après avoir visualisé le tableau il convient de préciser quelles cellules sont des en-têtes. Sur cette figure la cellule « mois » vient d'être définie par le webmestre comme une cellule d'en-tête puisqu'elle est colorée en vert. La balise `<TD>` de cette cellule va de ce fait être remplacée par la balise `<TH>`.

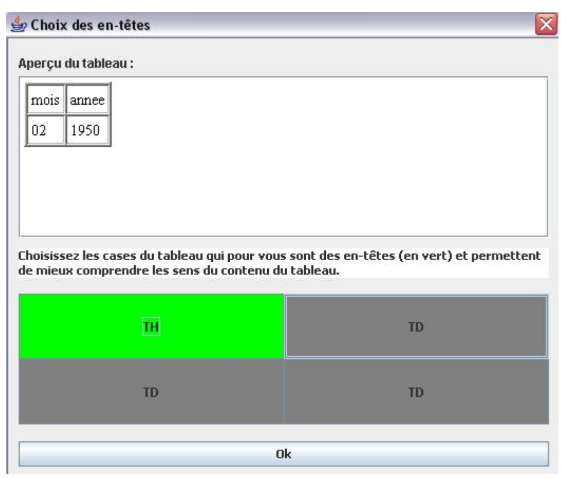


FIG. 6.7 – Interface du plugin de correction de la spécification 5.1.

6.2.5 Exemple d'application et résultats

Exemple d'application

Pour illustrer le travail de correction réalisé par notre plateforme de traitements de pages web, nous avons étudié le site web de la ville de Chambray-les-Tours. Une étude préliminaire [37] réalisée en 2006 avait permis de détecter de nombreuses erreurs concernant les images et les tableaux (concernant des attributs obsolètes utilisés dans des tableaux de mise en page). Dans cet exemple nous nous sommes intéressés à une page de ce site, a savoir la page « Loisirs et Culture » représentée par la figure 6.8.



FIG. 6.8 – Une page du site web de Chambray-les-Tours.

Dans le but d'évaluer la qualité des transformations effectuées par notre plateforme, nous avons testé cette page avec le validateur Ocawa, avant et après les transformations. Les résultats obtenus sur la page originale sont les suivants : 47 erreurs de niveau 1, 161 erreurs de niveau 2 et 1 erreur de niveau 3.

Tout d'abord, notre plateforme permet au webmestre d'ajouter des textes alternatifs aux images (figure 6.9). Puis, l'interface illustrée par la figure 6.6 permet de transformer les tableaux de mise en page utilisés pour les liens principaux dans le haut de la page : « La Ville », « Economie », « Chambray Pratique »... A ce moment nous avons choisi de supprimer les tableaux de mise en page et de garder en revanche la même disposition pour les informations qu'ils contenaient.

Le tableau 6.2 présente les résultats donnés par le validateur Ocawa, après modification de la page par notre correcteur. Nous pouvons entre autre remarquer que sur les 109 erreurs de niveau 2 concernant des attributs non supportés dans les balises, 56 ont été corrigées. Ces 56 erreurs correspondaient à des éléments non supportés dans la balise `<td>`. Or ces dernières étant utilisées pour des tableaux de mise en page, elles ont disparu avec la suppression du tableau au profit d'une mise en page avec des balises `<div>`. Finalement, la page d'accueil du site web de Chambray-les-Tours comportait au départ 209 erreurs ; après l'exécution de notre plateforme de traitement des pages web, cette même page ne comportait plus que 89 erreurs. Notons que ces 89 erreurs correspondent à des points de contrôle pour lesquels nous n'avons pas encore implémenté de plugins de correction.

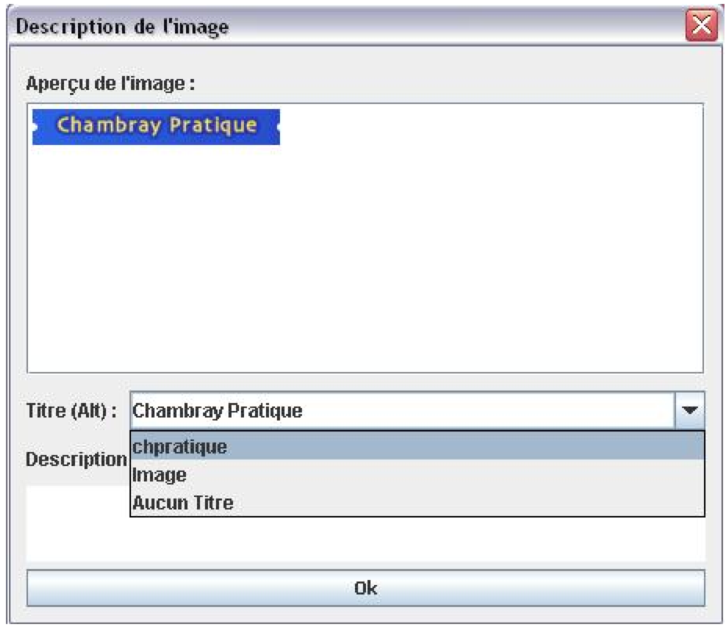


FIG. 6.9 – Interface du plugin permettant d'ajouter des textes alternatifs aux images (spécification 1.1).

Niveau des erreurs	Balises/attributs des erreurs détectées	Nombre d'erreurs avant correction	après correction	% de résolution
1	`<img>` alt	47	0	100%
2	`<a>` onfocus onblur	34	34	0%
2	`<img><td><hr>` attributs non supportés	109	53	51%
2	`<form>` label	1	1	0%
2	`<img>` alt (liens)	17	0	100%
3	lang	1	1	0%

TAB. 6.2 – Résultats d'Ocawa sur une page web avant et après modification par notre correcteur.

Résultats obtenus sur les sites web des communes d'Indre-et-Loire

Pour évaluer notre plateforme de correction des erreurs d'accessibilité, nous avons effectué les corrections concernant les spécifications 1.1, 5.3 et 5.5 des WCAG sur les sites web des 50 communes d'Indre-et-Loire listés dans le tableau B.1 de l'annexe B. L'annexe C donne les résultats de l'évaluation de ces sites web après passage du correcteur. Les figures 6.10 et 6.12 présentent respectivement, pour chaque directive, le nombre d'erreurs et de warnings, avant et après passage du correcteur. La figure 6.11 est identique à la figure 6.10, avec les histogrammes des directives 3 et 11 tronqués afin de mieux visualiser les résultats obtenus sur les autres directives.

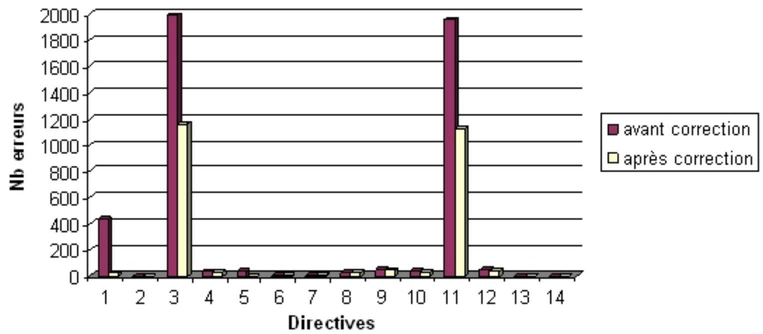


FIG. 6.10 – Nombre d'erreurs par directive avant et après passage du correcteur.

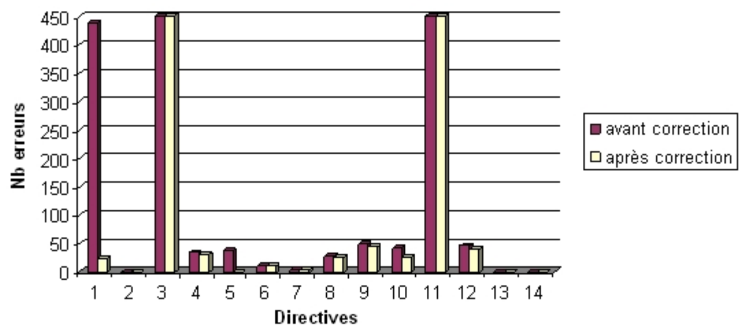


FIG. 6.11 – Nombre d'erreurs par directive avant et après passage du correcteur (avec directive 3 et 11 tronquées).

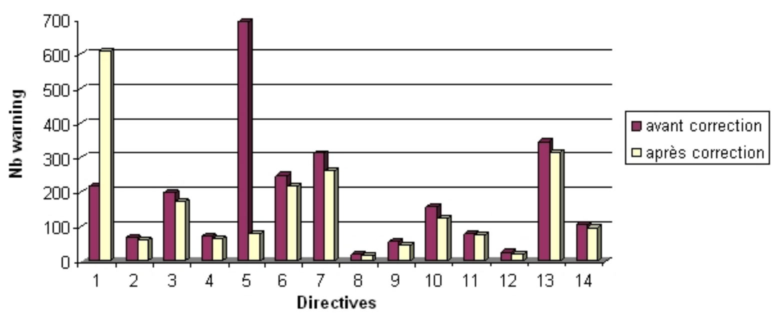


FIG. 6.12 – Nombre de warnings par directive avant et après passage du correcteur.

Dès lors, nous pouvons observer une diminution des erreurs (47% environ) et des warnings (17% environ) pour les directives que nous n'avons pas tenté de corriger. Nous avions précédemment observé ce phénomène sur l'exemple d'application. La correction de la spécification 5.3 — par rapport à la suppression des tableaux de mise en page— avait en même temps supprimée des erreurs concernant par exemple l'utilisation d'attributs obsolètes dans la balise `TABLE` et les attributs `TD` et `TR`.

En ce qui concerne les directives 1 et 5 pour lesquelles certaines spécifications ont été corrigées, nous remarquons une diminution du nombre d'erreurs : 415 erreurs corrigées pour la directive 1 et 37 erreurs corrigées (soit 100%) pour la directive 5. Il y a également une importante diminution du nombre de warnings pour la directive 5 (616 warnings corrigés), mais une forte augmentation du nombre de warnings pour la directive 1 (supplément de 392 warnings). Cela peut s'expliquer par le fait que les erreurs de cette directive portant sur l'absence de texte alternatif pour les images, se sont transformées en warnings conseillant de vérifier la pertinence du texte alternatif.

Les figures 6.13 et 6.14 représentent le nombre d'erreurs et de warnings concernant la directive 1 pour chaque site web évalué, respectivement avant et après correction. Nous voyons, come nous l'attendions, une forte diminution du nombre d'erreurs qui se sont transformées en warnings. Les figures 6.15 et 6.16 concernent quant à elles la directive 5 : nous observons une très nette diminution du nombre de warnings, et dans la majeure partie des cas, une suppression des warnings et des erreurs.

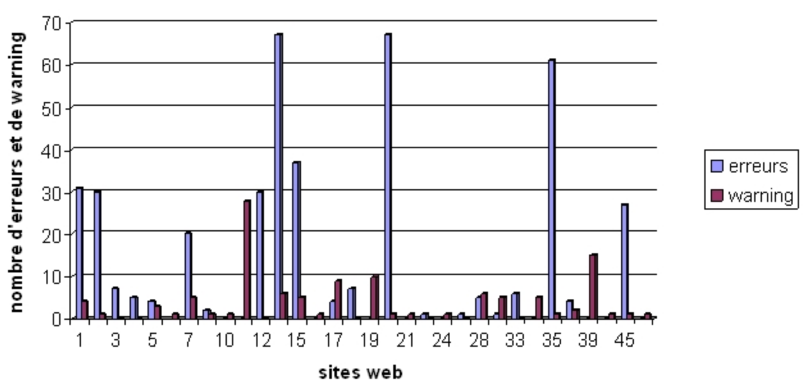


FIG. 6.13 – Nombre d'erreurs et de warnings concernant la directive 1 avant correction.

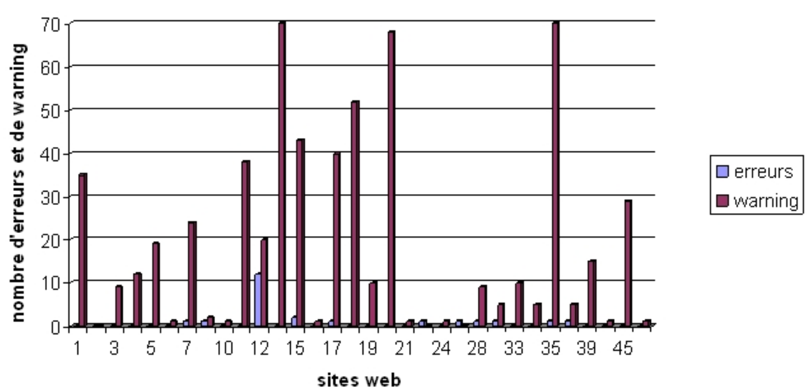


FIG. 6.14 – Nombre d'erreurs et de warnings concernant la directive 1 après correction.

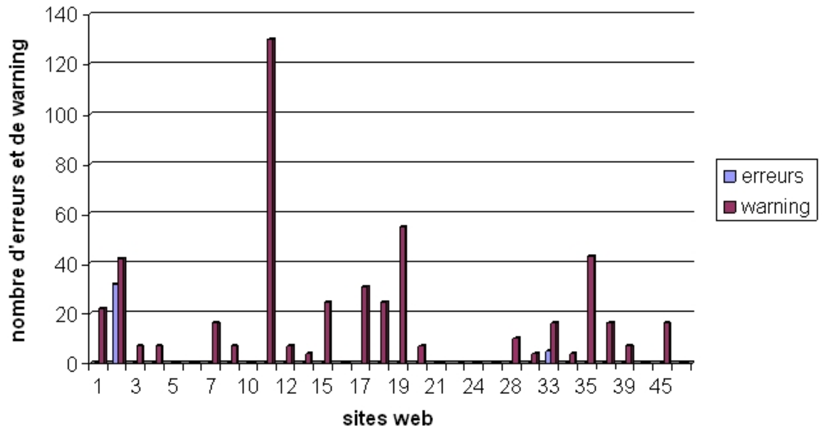


FIG. 6.15 – Nombre d'erreurs et de warnings concernant la directive 5 avant correction.

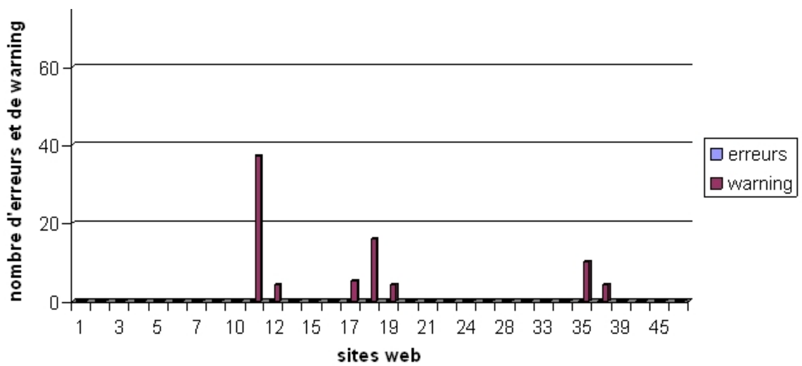


FIG. 6.16 – Nombre d'erreurs et de warnings concernant la directive 5 après correction.

Les spécifications implémentées montrent de bons résultats lors du passage au correcteur automatique. Reste qu'ils sont bien évidemment à modérer : le nombre de warnings ayant augmenté, le site web ne sera réellement accessible que si le webmestre a pris soin de renseigner correctement les informations demandées par notre correcteur.

6.3 Réorganisation de pages web

Les internautes voyants considèrent une page web comme un unique document. Cependant il faut bien admettre que ce document est structurellement composé de différentes zones distinctes et que sa lecture passe par la maîtrise d'un espace techniquement organisé (zones de contenu, de recherche, de navigation. . .). En termes d'ergonomie des sites web, nous avons rappelé dans la section 1.4.1 que cette structure géographique des différentes zones dans l'espace doit être cohérente d'une page à l'autre pour conserver une unité visuelle. Cette dernière permet effectivement à l'utilisateur d'accéder rapidement à la zone souhaitée sans avoir besoin de lire le document dans son ensemble. Finalement, si nous parlons couramment d'une « page » web — par référence à l'imprimé ou à la structure du code HTML —, ce n'est pourtant pas l'unité de mesure adoptée, que ce soit par les internautes ou par les webmestres.

En ce qui concerne les internautes non-voyants, l'utilisation de lecteur d'écran — couplé à une synthèse vocale ou à un clavier braille — implique une lecture linéaire de la page (section 1.2.3). La page web est alors considérée comme un unique document à lire dans sa totalité. Ce mode de lecture va à l'encontre de celui adopté par les internautes voyants, et pour cause : avec une page web comportant de nombreux éléments (logo, titre, sommaire, publicité. . .) dans le haut de la page, une translation simple dans un média séquentiel réduit dramatiquement l'accès au contenu pertinent. Un internaute non-voyant peut donc rencontrer de sérieuses difficultés pour accéder à l'information d'une page web, même si cette dernière respecte les critères d'accessibilité.

Illustrons ce problème en nous appuyant sur le site web de la mairie de Chambray-les-Tours, qui représente un des sites web pour lesquels nous avons étudié l'accessibilité et dont les résultats sont donnés dans l'annexe B. La figure 6.17 représente la page du site web de Chambray-les-Tours qui s'affiche lorsque l'internaute clique dans le menu (A) sur le lien « La ville ». Nous identifions visuellement différents blocs d'informations. Le bloc (A) joue le rôle de menu principal et le bloc (B) celui de sous-menu. Les blocs (C) et (D) correspondent respectivement à un menu propre à la page (et à ses sous-pages) ainsi qu'à une zone d'informations. Un internaute voyant peut accéder directement au contenu de la page qui l'intéresse (zone (C) ou (D)) tandis qu'un internaute non-voyant devra préalablement passer par la lecture des menus (A) et (B). Cela induit une perte de temps considérable lorsque, d'une part il est persuadé d'être sur la page internet qu'il cherche, et d'autre part il doit relire ces informations redondantes d'une page à l'autre du site web.

Pour accélérer la vitesse de lecture *via* un lecteur d'écran, nous avons réalisé un outil capable de détecter dans une page web différentes zones d'informations, tel que le ferait un internaute voyant. Dans un premier temps, nous différencions les zones de navigation de celles de contenu (section 6.3.3). Ainsi, lors de la visite d'une page d'un site web, l'internaute a la possibilité d'afficher l'une ou l'autre de ces zones — ou encore les deux zones, ce qui a pour conséquence de restituer la page originale. Dans un deuxième temps, nous devons préciser que tout ce qui n'est pas un menu n'est pas forcément de l'information utile propre à la page. Nous pouvons par exemple trouver dans un site web des bannières de publicité, le logo du site, un accès à un moteur de recherche. . . Tout ceci correspond généralement à des

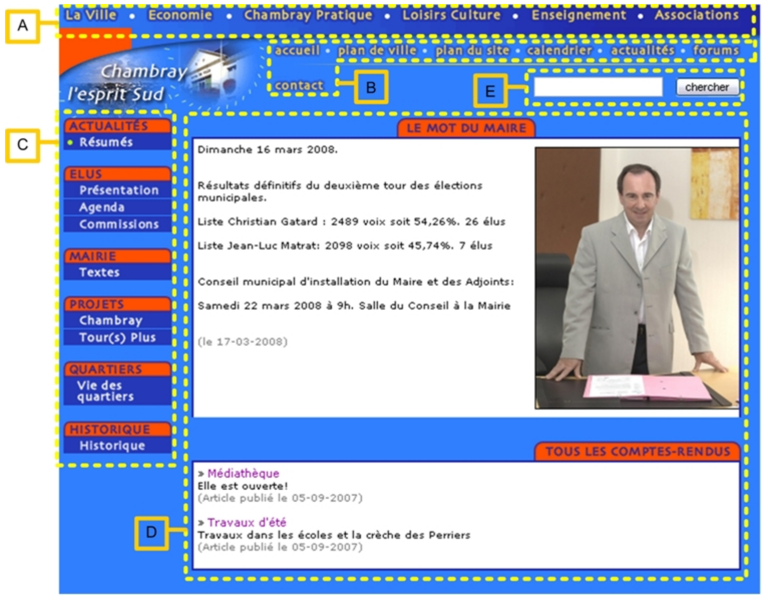


FIG. 6.17 – Exemple de site web avec ses différentes zones.

éléments qui se retrouvent sur toutes les pages du site mais qu'il n'est pas utile de relire à chaque nouvelle page. Nous proposons alors de détecter toutes les zones redondantes sur les pages du site, et ce dès la deuxième page visitée du site. Notons qu'une zone est considérée comme redondante dès lors qu'une zone identique a été repérée sur la page précédemment affichée par le navigateur (la redondance n'est pas identifiée sur la totalité du site web). Cette étape de détection des zones redondantes fait l'objet de la section 6.3.2.

Nous distinguons finalement quatre types de zones : les menus redondants, les menus spécifiques à la page, les zones redondantes qui ne sont pas des menus, et les zones d'information propre à la page qui ne sont pas des menus. L'utilisateur peut choisir d'afficher ou non chacune de ces zones. Notre outil propose également de réorganiser les éléments de la page pour permettre à l'internaute de personnaliser encore davantage son affichage ; ainsi il peut déterminer l'ordre d'apparition des zones qu'il a choisi d'afficher. Avant de décrire précisément les traitements de la page que nous venons d'exposer, une première étape consiste à préparer et à nettoyer le code source de la page web.

6.3.1 Construction de l'arbre représentatif de la page

Afin de préparer le code HTML pour la création de l'arbre, nous avons épuré ce code en fonction de nos besoins. Pour chaque page web que l'utilisateur souhaite visualiser, nous conservons uniquement les structures de bloc et les liens, c'est-à-dire les balises `<DIV>`, `<SPAN>` et `<A>`. Précisons qu'un titre ne doit pas être assimilé à un paragraphe et que deux paragraphes qui se suivent doivent rester distincts. De ce fait les éléments de titres `H1..H6` et les balises `<P>` représentant des paragraphes sont également conservés. De plus, les tableaux de données, les formulaires et les listes — c'est-à-dire les balises `<TABLE>`, `<FORM>`, `<UL>`,

<OL> et <DL> — sont considérés comme un bloc unique. Quant aux balises dont le contenu est vide, elles sont bien évidemment supprimées.

Une fois le code source de la page nettoyé, chaque page du site web est représentée sous la forme d'un arbre, chaque bloc présent dans la page HTML constituant un nœud de l'arbre. La figure 6.18 illustre deux exemples de pages HTML P_1 et P_2. Nous y distinguons des zones $z_{i,k}$ qui sont considérées comme des blocs utiles des pages k=1,2. La figure 6.19 illustre les deux arbres représentatifs correspondants aux pages P_1 et P_2.

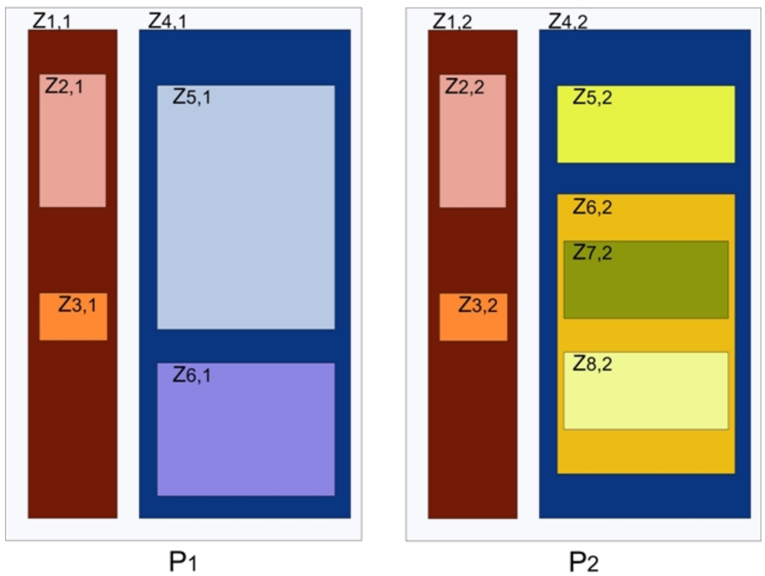


FIG. 6.18 – Schématisation de deux pages web d'un même site.

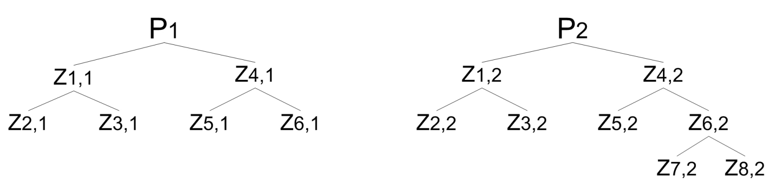


FIG. 6.19 – Arbres représentatifs des deux pages.

6.3.2 Détection des parties redondantes sur les pages

Lorsqu'une information est redondante sur plusieurs pages, nous considérons que c'est une information générale au site web. Ainsi, lorsqu'un utilisateur navigue sur un site web, ces informations redondantes peuvent être supprimées dès lors qu'elles ont déjà été visualisées sur une page précédente. Pour détecter deux zones identiques, nous utilisons la méthode des n-grammes (un n-gramme est une sous-séquence de n éléments construite à partir d'une

séquence donnée). Chaque zone peut alors être représentée par un vecteur contenant la fréquence d'apparition de chacun des n-grammes. Soit $z_{i,k}$ et $z_{j,l}$ deux zones respectivement sur les pages P_k et P_l. Les équations 6.1 et 6.2 permettent respectivement de définir deux zones strictement identiques et deux zones considérées comme identiques.

$$z_{i,k} = z_{j,l} \Leftrightarrow \cos(ngram(z_{i,k}), ngram(z_{j,l})) = 1 \tag{6.1}$$

$$z_{i,k} \approx z_{j,l} \Leftrightarrow \cos(ngram(z_{i,k}), ngram(z_{j,l})) > \theta \tag{6.2}$$

avec θ un seuil fixé expérimentalement. Par définition, le cosinus de l'angle formé entre les deux vecteurs $ngram(z_{i,k})$ et $ngram(z_{j,l})$ est le rapport du produit scalaire des vecteurs sur le produit des normes de chacun des vecteurs.

Lorsqu'une zone est considérée comme identique à une autre zone, elle est marquée en vue de sa suppression dans l'arbre représentatif de la page. Pour l'élimination des nœuds marqués, le parcours de l'arbre est effectué en largeur d'abord et le fils d'un nœud marqué est rattaché au père de ce dernier. Sur les deux pages de la figure 6.18, considérons que : $z_{1,1} = z_{1,2}$, $z_{2,1} = z_{2,2}$, $z_{3,1} = z_{3,2}$ et que $z_{4,1} \approx z_{4,2}$. La figure 6.20 reprend les arbres des pages P_1 et P_2 de la figure 6.19, dans lesquels les zones redondantes ont été supprimées.



Fig. 6.20 – Arbres représentatifs des deux pages sans les zones redondantes.

6.3.3 Détection des menus

Un internaute peut avoir besoin d'utiliser le menu — qu'il soit redondant ou non — d'un site web. Un menu est considéré comme une succession de liens. Dès que l'arbre contient une fratrie de nœuds correspondant à des balises `<A>`, cet ensemble de nœuds est considéré comme un menu.

Sur les sites web non accessibles, il est courant de voir des menus composés uniquement d'images ou bien mis en forme à l'aide d'un tableau. Dans le premier cas, les images représentant des liens sont supprimées. Le lien est alors reconstruit en utilisant le texte alternatif de l'image, ou bien, s'il est inexistant, l'adresse de la cible du lien. Dans le second cas, ces tableaux n'ont pas lieu d'être puisque les recommandations du W3C précisent qu'ils ne doivent être utilisés que pour présenter des données et non pour faire de la mise en page. Si de tels tableaux sont détectés — la méthode de détection est celle qui a été présentée dans la section 6.2.4 —, ils sont automatiquement supprimés et leur contenu est restructuré à l'aide de balises `<DIV>`. Les tableaux de données sont quant à eux conservés et restitués tels quels.

Sur les deux pages représentées sur la figure 6.18, considérons que : $z_{1,k}$, $z_{2,k}$, $z_{3,k}$ pour $k = 1, 2$ sont des menus du site web et que $z_{5,2}$ correspond à un menu spécifique à la page P_2. La figure 6.21 reprend les arbres des pages P_1 et P_2 de la figure 6.19, dont tous les menus ont été supprimés.

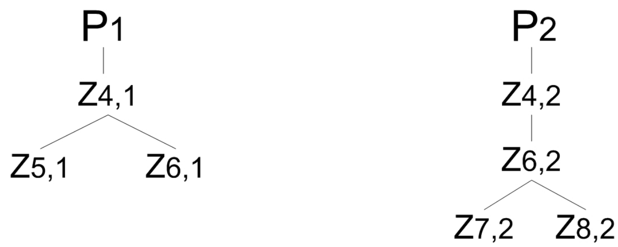


FIG. 6.21 – Arbres représentatifs des deux pages sans les menus.

6.3.4 Application et résultats expérimentaux

Valeur de θ

Pour fixer la valeur θ représentant le seuil qui permet de considérer deux zones comme identiques, nous avons réalisé une série de tests sur des sites web des Ministères de l'Intérieur, de la Santé, de l'Education Nationale et enfin de la Santé (ces sites web ayant été utilisés lors de l'étude e-mediacité [55] sur l'accessibilité des sites officiels). Nous avons comparé trois méthodes de mesure de similitude qui sont la méthode n-grammes, la distance de Levenshtein et celle de Jaro.

D'un côté, la distance de Levenshtein est égale au nombre minimal de caractères qu'il faut supprimer, insérer, ou remplacer pour passer d'une chaîne de caractères à l'autre. Ainsi plus deux chaînes sont similaires, plus la distance de Levenshtein est proche de zéro.

D'un autre côté, la distance de Jaro mesure la similarité entre deux chaînes de caractères selon le nombre de caractères correspondants dans les deux chaînes et le nombre de transpositions. Précisons que deux caractères sont considérés comme correspondants si la différence entre leurs positions dans leurs chaînes respectives ne dépasse pas la moitié de la longueur de la chaîne la plus longue ; les transpositions représentent quant à elles les interchangements entre caractères correspondants d'une chaîne pour obtenir l'autre chaîne. Cette distance est principalement utilisée pour la détection de similarité entre des chaînes courtes comme des mots de passe. Malgré le fait que nos chaînes à comparer soient composées de plusieurs mots, nous avons voulu tester cette distance, car de nombreuses chaînes — hormis les listes et les paragraphes — sont tout de même courtes, comme c'est le cas pour les liens.

Afin d'être comparées avec la méthode des n-grammes, les distances de Levenshtein et de Jaro-Winkler ont été normalisées pour obtenir une valeur entre 0 et 1 — le 0 représentant désormais l'absence de similarité.

Avec ces trois méthodes de similitude, à savoir les n-grammes, Lenvenshtein et Jaro, nous avons obtenu en moyenne respectivement les valeurs 0,01, 0,06 et 0,28 pour mesurer la similitude entre deux chaînes de caractères visuellement strictement différentes. La distance de Jaro donne les plus mauvais résultats, tandis que la méthode des n-grammes s'approche le plus de 0. Concernant la détection de zones strictement identiques, les trois méthodes sont équivalentes. En fixant $\theta = 1$, seules les zones strictement identiques seront reconnues comme redondantes par notre programme. Afin de considérer comme redondants des nœuds très similaires visuellement mais ayant des codes très légèrement différents (comme par exemple une différence de saut de ligne ou d'espace), nous avons fixé par expérience $\theta = 0,94$ avec la méthode des n-grammes. Ainsi, lorsqu'une différence inférieure à 6% entre deux nœuds est détectée, ces derniers sont considérés comme identiques.

La figure 6.22 illustre le menu du site de l'Education Nationale avec deux configurations possibles : d'une part lorsque l'internaute visualise la page « L'école dans votre région » (a) et d'autre part lorsqu'il visite la page « Le système éducatif » (b). Deux sous-menus différents s'affichent alors, ce qui induit trop de différences entre ces deux menus pour que ceux-ci puissent être considérés comme identiques (différences supérieures à 6%).



(a) (b)

FIG. 6.22 – Site du Ministère de l'Education Nationale : avec le sous-menu « L'école dans votre région » (a) et avec le sous-menu « Le système éducatif » (b).

Néanmoins certains utilisateurs pourraient au contraire penser que ce sont deux menus identiques, avec une variante, et qu'ils devraient de ce fait être considérés comme redondants lors de la navigation. Dans ce cas la valeur de $\theta = 0,94$ est bien trop élevée — dans cet exemple les deux menus sont identiques à 77%. Avec $\theta = 0,70$, ces deux menus auraient été considérés par notre programme comme identiques. Cette appréciation de la ressemblance entre deux zones étant subjective, nous avons choisi de laisser l'utilisateur paramétrer lui-même le pourcentage au-delà duquel deux zones sont considérées comme identiques (la valeur par défaut étant fixée à 94%).

Exemple d'application

Dans cette section, nous présentons un exemple d'application sur le site web de Chambray-les-Tours. Précisons qu'ici l'internaute désire n'avoir accès qu'aux informations propres à la page web et qui ne sont pas des menus. Toutes les zones redondantes (menu ou non) sont supprimées. Considérons que l'internaute décide de consulter la page présentée par la figure 6.17. Comme c'est la première page du site qu'il visite, aucune zone n'est considérée comme redondante. L'utilisateur a de plus précisé qu'il ne souhaitait pas afficher les menus ; par conséquent les zones (A), (B) et (C) sont supprimées. La figure 6.23 représente la page qui s'affiche dans son navigateur par défaut.

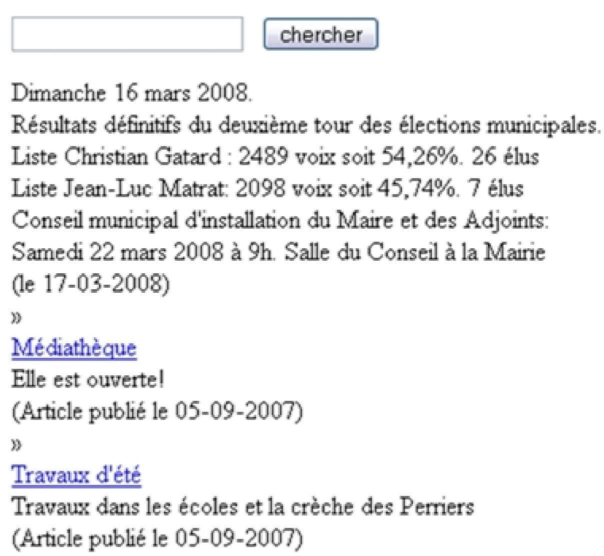


FIG. 6.23 – Affichage du contenu propre à la page « La ville ».

La figure 6.24 représente une autre page de ce site web vers laquelle l'internaute se dirige ensuite. Nous y trouvons un menu (A'), un sous-menu (B') ainsi qu'une zone de recherche (E') identiques respectivement aux zones (A), (B) et (E) de la page de la figure 6.17. Ces trois zones sont alors supprimées. La zone (C') est quant à elle détectée comme un menu (non redondant) et sera donc également éliminée.



FIG. 6.24 – Page « Loisirs » du site web de Chambray-les-Tours.

Finalement l'internaute n'a accès qu'au texte de la zone (D') représentant le contenu propre à cette page et n'étant pas un menu (figure 6.25).

Politique culturelle de la ville
La ville de Chambray-Lès-Tours propose dans le cadre de sa politique culturelle une programmation offrant à la population des formes artistiques les plus variées. Parallèlement, la ville de Chambray-Lès-Tours développe des actions culturelles en collaboration avec les établissements scolaires et propose une politique de tarifs permettant l'accès au spectacle du public le plus large. Des partenariats nombreux se développent avec les institutions et les associations chambraisiennes ainsi qu'avec les acteurs culturels de l'agglomération tourangelle. Le service culturel de la ville a pour objectif d'offrir au public une programmation cohérente, en concertation avec l'école de musique et la bibliothèque.

FIG. 6.25 – Affichage du contenu propre à la page « Loisirs ».

Rappelons tout de même que la suppression des zones n'est pas une obligation. Les menus, même redondants, peuvent permettre de se rediriger par la suite vers une autre page. L'utilisateur aurait alors pu choisir d'afficher par exemple tout d'abord les menus et contenus propres à la page, puis les menus redondants et enfin les contenus redondants.

6.4 Conclusion

La plateforme de transformation présentée dans la première partie de ce chapitre peut être utilisée en amont par le webmestre pour corriger ses pages, lequel peut choisir de les transformer de manière automatique ou manuelle. Précisons que le mode automatique constitue une version diminuée du correcteur en mode manuel puisque, d'une part certaines erreurs ne peuvent pas être corrigées automatiquement, et d'autre part, celles pour lesquelles nous autorisons une correction automatique ne seront pas forcément corrigées totalement. De ce fait, l'intérêt majeur de ce correcteur ne réside pas dans le mode automatique mais dans le mode permettant d'assister le webmestre. De plus, ce correcteur étant destiné aux webmestres débutants, nous avons fait en sorte que l'intervention du webmestre en mode manuel se fasse à l'aide d'interfaces s'utilisant de manière simple et intuitive.

Cette plateforme peut également être utilisée en aval par l'internaute. Dans ce cas, les pages sont modifiées à la demande de l'internaute avant visualisation sur le navigateur web. Seul le mode automatique est alors réalisable. Ce dernier permet alors d'améliorer (autant que possible en mode automatique) le niveau d'accessibilité des pages, et ce de manière transparente pour l'internaute.

A terme, d'autres traitements seront développés pour permettre la correction d'un plus grand nombre d'erreurs WCAG. Et, cette plateforme pourra également être utilisée directement à la suite du validateur décelant les erreurs des pages web.

Toujours dans le but de transformer les pages web — mais cette fois uniquement pour les internautes —, nous avons présenté dans la deuxième partie de ce chapitre un outil d'extraction d'informations. Celui-ci permet à un internaute utilisant un lecteur d'écran d'accéder plus rapidement au contenu de la page qui l'intéresse. L'utilisateur peut également réorganiser les différents éléments de la page qu'il a décidé d'afficher.

Actuellement nous distinguons quatre zones d'informations, à savoir les menus redondants ou non et les zones de contenu redondantes ou non, mais nous pouvons envisager d'en détecter d'autres comme par exemple celles correspondant à un moteur de recherche.

Les travaux présentés dans ce chapitre ont donné lieu à l'encadrement de projets de fin d'études au Département d'Informatique (DI) de Polytech'Tours [10][18] et ont conduit à des publications [41][40][32].

Conclusion

Grâce à l'adoption des aides techniques, les déficiences physiques ne constituent plus un obstacle à l'utilisation de l'informatique. Les personnes handicapées ont désormais la possibilité d'accomplir des tâches informatiques au même titre que les personnes valides — chose qui était quasiment impossible par le passé. Néanmoins, dans le monde du travail, les personnes handicapées ont longtemps été éconduites. Or ce n'est pas la personne elle-même qui est incapable de s'adapter à son environnement à cause de son handicap mais bien l'environnement en question qui l'exclut à cause de son handicap et des différences que ce dernier engendre.

Contre cette exclusion, Internet constitue un gigantesque espace de rassemblement pour les individus ayant des intérêts ou des besoins communs. Internet se présente alors comme un outil capable de former de petites communautés de personnes dispersées à travers le monde et « partageant » un même handicap. Par ce biais, les individus peuvent communiquer entre eux et ainsi s'inscrire au sein d'un espace virtuel difficilement transposable dans la réalité.

De plus, toujours pour combattre les exclusions, Internet apparaît comme un instrument qui offre aux personnes handicapées un accès à l'information — par exemple en proposant aux personnes malvoyantes une alternative aux documents imprimés. Il constitue également un moyen de réaliser plus facilement des tâches administratives — déclaration d'impôt, demande d'un acte de naissance... — ou de la vie courante — achats, réservation de billets de train, tenue des comptes bancaires...

Toutefois, pour que les portes de ce monde virtuel soient ouvertes à tous sans distinction et que chacun puisse en profiter pleinement, de nombreuses avancées restent à réaliser concernant l'accessibilité du web. Cette bataille, que nous souhaitions vivement mener, se livre simultanément sur deux grands fronts à savoir l'adaptation « au » web (ou adaptation de l'utilisateur) ainsi que l'adaptation « du » web (ou adaptation du contenu).

L'adaptation de l'utilisateur

L'adaptation de l'utilisateur au web consiste à proposer aux personnes handicapées des dispositifs techniques afin qu'elles puissent bénéficier d'Internet. L'accès à une information doit pouvoir s'effectuer *via* différents médias : par exemple le contenu d'un site Internet et une messagerie doivent pouvoir être consultés respectivement par l'intermédiaire de l'écran réduit d'un ordinateur de poche et depuis une boîte vocale. Le problème est similaire pour

les personnes handicapées qui accèdent à l'information par des médias spécifiques dépendant de leur capacité. Rappelons que certains médias peuvent être coopératifs, comme la synthèse vocale (pour une prise de connaissance rapide du texte) et le clavier braille (qui offre une plus grande précision notamment en ce qui concerne l'orthographe).

Nous avons sur ce point réalisé différents outils permettant de faciliter l'accès au web avec différentes aides techniques : le clavier virtuel, le générateur de plan ou encore le traducteur en braille abrégé s'inscrivent dans ce cas. La figure 6.26 permet de résumer les caractéristiques principales de chaque outil que nous avons réalisé pour les internautes. Si la plupart sont destinés aux internautes déficients visuels, nous avons néanmoins élargi la population ciblée aux personnes handicapées des membres supérieurs ainsi qu'aux personnes non atteintes de handicaps.

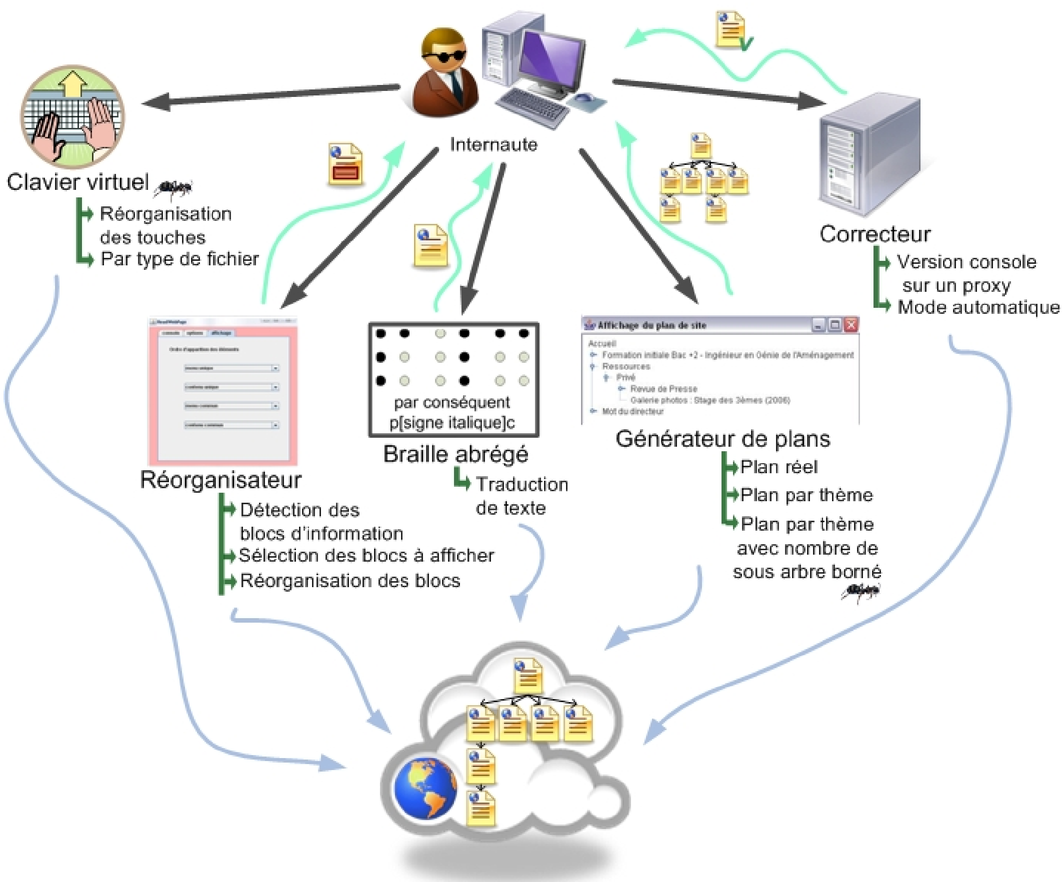


Fig. 6.26 – Les outils au service de l'internaute.

Rappelons brièvement le rôle et le fonctionnement de ces outils :

- Nous avons utilisé un algorithme de fourmis artificielles pour construire un clavier virtuel adapté au type de texte que doit saisir une personne handicapée physiquement. Grâce à celui-ci, l'organisation des touches est optimisée pour réduire les déplacements lors de la saisie de textes de même type que celui ayant servi de modèle pour la

construction du clavier. L'utilisateur voit ainsi sa fatigue réduite par exemple lors de la saisie de formulaire d'inscription, de l'envoi de mails *via* un formulaire, et aussi pour toutes ses tâches bureautiques courantes. Précisons que ce type de clavier peut également profiter aux personnes utilisant des moyens de communication nomade où la saisie avec un stylet est imposée.

- Les lecteurs d'écran (couplés à une synthèse vocale et/ou à un clavier braille) produisent une lecture linéaire de la page aux personnes handicapées visuellement. Pour pallier au problème de relecture des informations redondantes d'une page à l'autre d'un même site web, nous avons réalisé un réorganisateur de pages web qui permet de choisir les blocs d'information à afficher ainsi que leur ordre. Pour accélérer encore davantage la lecture de l'information pertinente de la page, nous proposons un traducteur en braille abrégé.
- Nous nous sommes intéressés aux problèmes que peuvent rencontrer les internautes pour se repérer et se diriger au sein d'un site web lorsque celui ne possède pas de plan ou que ce dernier est trop dense. Nous avons pour cela proposé un générateur de plan de site web avec un affichage sous forme de liste. A l'instar de son homologue destiné aux webmestres, l'utilisateur peut choisir de générer le plan réel du site, la plan par thème, ou celui par thème avec un nombre de sous-arbre borné à chaque niveau du plan pour s'adapter à ses capacités de mémorisation s'il est non-voyant.
- Finalement, en attendant la mise en application totale des normes d'accessibilité par les sites web, nous proposons à l'internaute un correcteur de page web chargé d'effectuer les modifications réalisables automatiquement. Ce correcteur propose par exemple de résoudre les problèmes liés aux tableaux de mise-en-page dans lesquels la lecture linéaire n'est pas possible.

L'adaptation du contenu du web

L'adaptation du contenu du web relève de la responsabilité des créateurs de contenu pour le web. Ils ont à leur charge différents points à respecter. Tout d'abord l'accessibilité de l'information d'une page dépend bien évidemment des possibilités offertes par le lecteur d'écran, mais aussi du contenu et de la disposition de la page. Nous trouvons de nombreuses manières de proposer l'information sur une page que ce soit textuellement, avec des images, des vidéos ou des bandes sonores, et encore diverses façons de présenter cette information avec des frames ou des tableaux par exemple. Reste que le meilleur lecteur d'écran ne peut s'adapter aux différents cas et ne possède pas toujours tous les renseignements nécessaires pour restituer parfaitement l'information. Par conséquent de nombreux efforts sont à la charge du webmestre qui doit alors fournir un site web accessible, c'est-à-dire respectant les normes établies. Ensuite, des problèmes d'ergonomie des sites peuvent également rendre l'accès à l'information plus difficile pour les personnes handicapées. Par exemple lorsqu'un utilisateur visite un site Internet, il se construit une carte mentale des pages qu'il a déjà visitées. Sans cette représentation mentale, ou s'il n'arrive pas rapidement à l'acquérir, il lui est impossible de se diriger et dans ce cas il se perd dans le labyrinthe composé par les liens et les pages pour finalement abandonner sa visite du site. La structure du site doit donc être claire, spécialement pour les personnes aveugles pour lesquelles l'acquisition de cette

représentation est plus difficile. En l'occurrence, il est du ressort du webmestre de structurer son site web pour que les regroupements des pages, mais aussi les intitulés de celles-ci, soient représentatifs de leur contenu. Pour terminer, l'adaptation de l'information est également un point crucial de l'adaptation du web. Il s'avère utile de réfléchir à l'organisation de la page en pensant par exemple à la rapidité d'accès à l'information pertinente pour une personne non-voyante — ou plus généralement pour une personne qui utilise un dispositif ne lui permettant pas de lire la page en diagonale.

Nous avons également contribué à favoriser l'adaptation du web en proposant par exemple des outils pour aider les webmestres à rendre leur site accessible au regard des normes (guide, validateur, correcteur) et pour les épauler lors de la réalisation de la structure globale de leur site. La figure 6.27 permet de résumer les caractéristiques principales des outils auxquels peut accéder le webmestre dans sa démarche d'accessibilité.

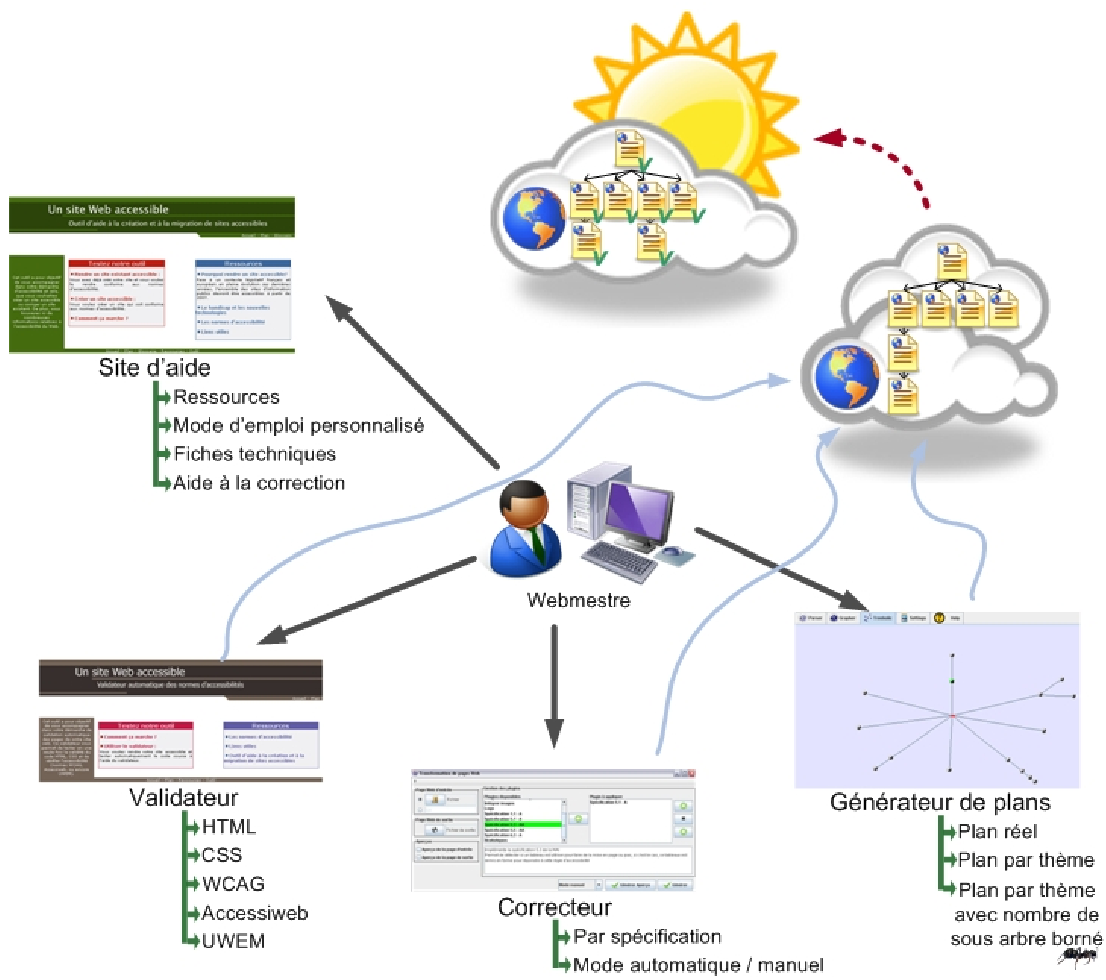


FIG. 6.27 – Les outils à destination du webmestre.

Rappelons brièvement le rôle et le fonctionnement de ces outils destinés aux webmestres :

– Le manque d'information sur l'accessibilité constitue un premier frein dans la démarche

d'un webmestre — le deuxième frein correspondant aux difficultés de compréhension des règles d'accessibilité et à la manière de les mettre en œuvre. C'est pour cette raison que nous avons créé un outil d'aide contenant des informations générales sur l'accessibilité, des fiches technique, une aide à la correction des erreurs fréquemment rencontrées, ainsi qu'une démarche personnalisée pour rendre ou créer un site accessible.

- Lors de sa démarche, le webmestre doit utiliser des validateurs pour tester les normes HTML, CSS et celles d'accessibilité. Pour ce faire nous avons élaboré un unique validateur capable de tester toutes ces normes ainsi que les normes Accessiweb et UWEM. Sa fonctionnalité multi-pages et son système d'affichage par type d'erreur s'adapte parfaitement aux webmestres débutants.
- Une fois les erreurs détectées, la phase de correction peut débuter. Nous avons à ce propos conçu un correcteur des normes d'accessibilité modifiant automatiquement le fichier source d'après les informations fournies par le webmestre *via* les interfaces. Précisons qu'à ce jour, toutes les spécifications n'ont pas été développées, ce qui implique que la seule utilisation de notre correcteur ne permet pas au site web de respecter toutes les erreurs décelées par le validateur.
- Finalement, pour aller encore plus loin, le webmestre peut s'intéresser à la structure de son site web pour qu'il présente une arborescence de pages équilibrée et thématique. Sur ce point, nous nous sommes intéressés à la capacité des fourmis à fournir un arbre de poids minimum à partir d'un graphe représentant la structure réelle du site et pondéré selon la distance (similarité) entre les pages. Le webmestre peut choisir d'afficher (sous forme graphique ou de liste) la structure réelle de son site, le plan par thème, ou encore par thème avec un nombre maximum de sous-arbres à chaque niveau de l'arborescence. Rappelons que la version actuelle de notre générateur utilise, pour les générations par thème, une mesure de similarité basée sur un TF/IDF, qui pour présenter un intérêt dans une utilisation concrète, devra être remplacée par une autre plus performante.

Perspectives

Les perspectives sont nombreuses concernant ce domaine en constante évolution ; elles ont été évoquées en fin de chaque chapitre. Elles concernent essentiellement la finalisation des interfaces avec les utilisateurs, qu'ils soient webmestres ou utilisateurs handicapés, afin de profiter de leurs remarques et expériences, notamment pour proposer diverses possibilités d'utilisation des outils (que ce soit en local, *via* un proxy ou encore à la suite d'une autre entité). Nous pouvons effectivement remarquer que certains outils pourraient être mis en relation avec d'autres pour construire un système global. La figure 6.28 représente tous les outils réalisés — selon leur localisation géographique et le type de personnes à qui ils sont destinés — ainsi que les relations qui peuvent s'établir entre eux. Pour l'internaute, le système de correction automatique — si on s'en sert *via* un proxy comme illustré sur la figure 6.28 — pourrait être exécuté automatiquement après le validateur afin de ne corriger

que les spécifications nécessaires. De plus, le traducteur de braille abrégé pourrait être utilisé par l'outil de réorganisation pour convertir en braille abrégé les blocs d'informations que l'internaute a décidé d'afficher, ce pour une lecture encore plus rapide. En ce qui concerne le webmestre, des liens pourraient être ajoutés entre le validateur et les indications fournies par l'outil d'aide à l'accessibilité pour aider à corriger les principales erreurs. Et, comme pour l'internaute, le validateur devrait être capable de fournir au correcteur la liste des plugins à exécuter. D'autres perspectives sont à envisager pour que chaque entité soit utilisable et efficace dans des cas concrets. C'est le cas du correcteur de page pour lequel l'ensemble des spécifications devront être implémentées, du réorganisateur de page pour lequel un plus grand nombre de zones d'informations pourrait être détecté, ou encore du générateur de plan de site web pour lequel une mesure de similarité compétente (capable de détecter un thème commun même avec des termes différents) entre les pages devra être établie. En définitive, aux vues des évolutions récentes concernant les normes d'accessibilité, le validateur et le correcteur devront eux aussi évoluer. Ils devront tout d'abord s'adapter aux recommandations WCAG 2.0 qui remplaceront les normes WCAG 1.0. Ensuite le validateur multi-recommandations pourra également intégrer de nouvelles normes de pays européens dès que ces derniers auront accepté le schéma de migration vers les normes européennes UWEM.

Finalement, c'est en travaillant sur ces deux aspects, à savoir l'accessibilité au web et l'accessibilité du web, que des résultats significatifs en terme global d'accessibilité peuvent être obtenus. Si l'adaptation du web semble primordiale, la mise en place de sites web accessibles d'après les normes établies risque d'être longue. Prenons pour exemple la présence de plus en plus forte de nouvelles technologies « visuelles » sur le web comme la vidéo ou le flash. La mise en conformité de ces technologies, c'est-à-dire leur retranscription en équivalent textuel pour les personnes qui utilisent un lecteur d'écran, représente un travail considérable pour le webmestre qui pourra difficilement restituer toute l'information. Mais d'un autre côté, il n'est pas non plus envisageable d'empêcher l'utilisation de nouvelles technologies sur les sites web. L'adaptation au web, c'est-à-dire de l'utilisateur, s'avère donc un aspect à développer pour proposer aux internautes des logiciels adaptés à leurs handicaps.

Certes nous sommes loin de prétendre avoir traité toute l'étendue et la complexité de ces deux points. Il est vrai que nombreuses perspectives sont encore en suspens à la suite de nos travaux, et un vaste éventail de thèmes reste à aborder (par exemple l'amélioration d'aides techniques comme les systèmes alternatifs à la souris). La tâche est immense et l'objectif d'un web accessible n'est à ce jour envisageable que partiellement (pour les sites web publics). Force est de constater que les travaux en faveur de l'accessibilité sont encore à poursuivre et ce sans relâche. Reste que, nous ne pouvons qu'apprécier les nombreuses recherches et avancées — notamment politiques — qui montrent la prise de conscience d'un public de plus en plus large aux problèmes que peuvent rencontrer les personnes handicapées lors de l'usage d'Internet. Cette compréhension et cette prise en compte par la multitude nous laissent dans l'agréable espérance d'une poursuite des travaux commencés et de progrès imminents.

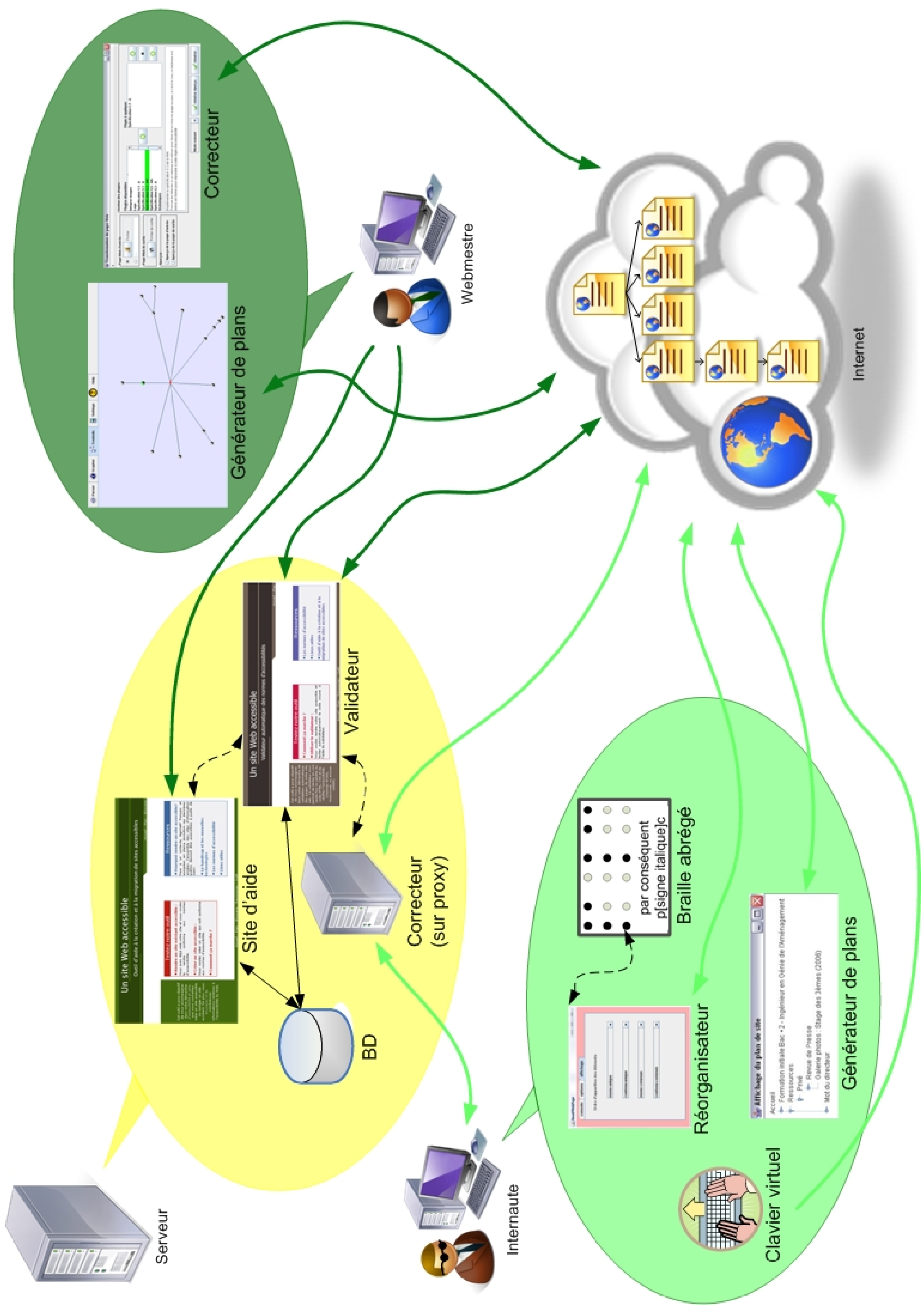


FIG. 6.28 – Représentation de l'ensemble des outils réalisés.

Annexe A

Algorithmes principaux utilisés lors de la transcription en braille abrégé

Un texte en braille abrégé est composé de mots — contenant une ou plusieurs contractions — de symboles, et de mots en intégral.

L'algorithme 5 présente le principe de la transcription d'un texte en braille abrégé. Il permet essentiellement de détecter les locutions (à l'aide d'un dictionnaire) et de retourner l'abréviation correspondante (algorithme 5 ligne 7). Si aucune locution n'est détectée (algorithme 5 ligne 9), l'étude est réalisée mot par mot (algorithme 6).

Algorithme 5 Algorithme principal de traduction en braille abrégé

Entrées: s : chaîne de caractères représentant le texte à convertir en braille abrégé
Sorties: sabrege : chaîne de caractères représentant s en braille abrégé

```
1:  sabrege ← « »
2:  Découper s en une suite de mots
3:  Prendre le 1er mot
4:  Tantque il reste des mots à traiter Faire
5:     motabrege ← « »
6:     Si le mot (et les suivants) représente une locution Alors
7:        motabrege ← transcription(locution)
8:     Sinon
9:        motabrege ← decoupe(mot)
10:    Finsi
11:    sabrege ← sabrege + motabrege
12: Fin tantque
13: retourner  sabrege
```

L'algorithme 6 analyse chaque mot pour trouver l'abréviation contenant le plus de caractères possibles. Lors de la recherche de l'abréviation (algorithme 6 ligne 4, algorithme 7), nous essayons de trouver soit un symbole si le mot dont on cherche l'abréviation est entier, soit une contraction s'il s'agit d'une séquence de caractères appartenant à un mot.

Algorithme 6 Découpe d'un mot pour rechercher une abréviation

ENTRÉES: mot : chaîne de caractères représentant le mot à abréger
SORTIES: motabrege : chaîne de caractères représentant mot en braille abrégé

motabrege ← « »
Tantque tous les caractères de mot ne sont pas abrégés **Faire**
  seqabrege ← « »
  seqabrege ← abreviation(mot)
  Si seqabrege ≠ « » **Alors**
    motabrege ← motabrege + seqabrege
    mot ← partie non encore abrégé de mot
  Sinon
    mot ← mot - (dernier caractère de mot)
  Finsi
Fin tantque
retourner motabrege

Algorithme 7 Recherche de l'abréviation

ENTRÉES: s : chaîne de caractères représentant le mot à abréger ou une partie d'un mot
SORTIES: sabrege : chaîne de caractères représentant mot en braille abrégé

Si longeur(s)=1 **Alors**
  retourner sabrege
Finsi
Si s est un mot complet **Alors**
  sabrege ← Recherche d'un symbole
Finsi
Si sabrege =« » **Alors**
  sabrege ← Recherche d'une contraction
Finsi
retourner sabrege

La ligne 8 de l'algorithme 7 permet de chercher une contraction pour un ensemble de caractères contenus dans un mot. Rappelons que lorsqu'une abréviation d'un ensemble de lettres contenu dans un mot est possible d'après les règles de contraction, il faut vérifier que cet ensemble de lettres appartient à une seule syllabe. Dans le cas contraire, l'abréviation ne doit pas se faire. De ce fait, lors de la traduction d'un mot en braille abrégé, il est nécessaire de distinguer les différentes syllabes d'un mot. L'algorithme 8 représente la méthode de séparation des mots en syllabes donnée dans la section 4.3, avec la fonction *est_exception*(a, b) permettant de détecter si les deux caractères a et b font partis de la liste des caractères inséparables.

Algorithme 8 Découpage en syllable

Entrées: mot : chaine de caractère
  $i \leftarrow 0$
  Tantque $i \neq$ longueur(mot) **Faire**
    Si est_voyelle(mot[i]) and est_consonne(mot[i+1]) and est_consonne(mot[i+2]) and est_voyelle(mot[i+3]) **Alors**
      Si est_exception(mot[i+1]+mot[i+2]) **Alors**
        mot ← souschaine(mot, 0, i+1) + / + souschaine (mot, i+1){V/CCV}
      Sinon
        mot ← souschaine(mot, 0, i+2) + / + souschaine (mot, i+2){VC/CV}
      Finsi
    Sinon Si est_voyelle(mot[i]) and est_consonne(mot[i+1]) and est_voyelle(mot[i+2]) **Alors**
      mot ← souschaine(mot, 0, i+1) + / + souschaine (mot, i+1){V/CV}
    Sinon Si est_consonne(mot[i]) and est_consonne(mot[i+1]) and est_consonne(mot[i+2]) **Alors**
      Si est_exception(mot[i+1]+mot[i+2]) **Alors**
        mot ← souschaine(mot, 0, i+1) + / + souschaine (mot, i+1){C/CC}
      Sinon
        mot ← souschaine(mot, 0, i+2) + / + souschaine (mot, i+2){CC/C}
      Finsi
    Sinon
      $i \leftarrow i + 1$
    Finsi
  Fin tantque
  retourner mot

Annexe B

Evaluation de l'accessibilité des sites web publics d'Indre-et-Loire

Nos travaux, destinés à faciliter la mise en accessibilité des sites web, ont initialement été créés pour les webmestres des collectivités locales d'Indre-et-Loire (37). Nous avons alors évalué l'accessibilité de sites web d'un échantillon aléatoire de 50 communes d'Indre-et-Loire. Le tableau B.1 représente les 50 communes sélectionnées pour cette étude. Pour indication, le nombre d'habitants (en 2005) de chaque commune est précisé. Pour certaines communes (essentiellement celles ayant le moins d'habitants) aucun site officiel n'a été trouvé : 71% des communes de moins de 2000 habitants ne possèdent pas de sites web, contre 31% pour les communes d'entre 2000 et 4000 habitants et 0% pour celles de plus de 4000 habitants. Finalement 35 sites web ont été évalués.

A savoir, les tableaux B.2 à B.7 représentent les résultats de chaque site web évalué par notre validateur (section 5.2 page 127). Les sites web sont identifiés par leur numéro dans le tableau B.1, et le nombre d'erreurs ou de warnings est indiqué par directive WCAG. Le tableau B.8 dresse le nombre d'erreurs et de warnings détectés par directive.

N°	Ville	Nb hab.	URL
1	Joué-lès-Tours	36000	http://195.154.219.130/scriptes_joue/default.xom
2	Saint Cyr /Loire	16300	http://www.saint-cyr-sur-loire.com/
3	Saint Pierre des Corps	15400	http://www.ville-saint-pierre-des-corps.fr/
4	Saint Avertin	14100	http://www.ville-saint-avertin.fr/
5	Amboise	12400	http://www.ville-amboise.fr/
6	Chambray les Tours	10600	http://www.ville-chambray-les-tours.fr/
7	Montlouis/ Loire	10380	http://www.ville-montlouis-loire.fr/php/accueil/accueil.php
8	Fondette	9980	http://www.ville-fondettes.fr/
9	Chinon	8170	http://www.ville-chinon.com/default.php
10	Ballan-Miré	7600	http://www.mairie-ballan-mire.fr/francais/accueil.htm
11	Monts	6950	http://www.monts.org/index.php
12	Loches	6370	http://www.ville-loches.fr/
13	Veignés	5470	http://www.veigne.fr/
14	Château-Renault	5200	http://www.ville-chateau-renault.fr/
15	Bléré	5020	http://www.blere-touraine.com/default.php
16	Luynes	4940	http://www.luynes.fr/
17	La Ville-aux-Dames	4520	http://www.ville-aux-dames.com/site/index.php
18	Veretz	4430	http://www.veretz.com
19	Esvres	4400	http://www.ville-esvres.fr/
20	Descartes	4020	http://www.ville-descartes.fr/
21	Langeais	3860	http://www.langeais.fr/sommaire.html
22	Monnaie	3830	http://www.ville-monnaie.fr
23	Nazelles-Négron	3630	*
24	Notre-Dame-d'Oé	3407	http://www.ville-de-notre-dame-doe.fr/
25	Monnaie	3300	http://www.ville-monnaie.fr/

Suite page suivante ...

N°	Ville	Nb hab.	URL
26	Vouvray	3110	`http://www.vouvray.org`
27	Azay-le-Rideau	3100	*
28	Savonnières	3020	`http://www.savonnieres.fr/`
29	Rochecorbon	2980	`http://www.mairie-rochecorbon.fr`
30	Chanceaux-sur-Choisille	2821	*
31	Cinq-Mars la Pile	2620	*
32	Artannes-sur-Indre	2500	*
33	Ligueil	2180	`http://ligueil.free.fr`
34	Richelieu	2160	`http://www.ville-richelieu.fr`
35	Larçay	2040	`http://www.ville-larcay.fr/actualite.htm`
36	Athée-sur-Cher	2000	`http://www.athee-sur-cher.fr/`
37	Neuillé-Pont-Pierre	1930	*
38	Avoine	1840	*
39	Château-la-Vallière	1530	`http://www.chateaulavalliere.com/index2.htm`
40	Savigny-en-Véron	1430	`http://savignyenveron.free.fr/`
41	Ambillou	1310	*
42	Savigné-sur-Lathan	1260	*
43	Neuvy-le-Roi	1160	*
44	Abilly	1100	*
45	Villandry	1080	`http://www.villandry.fr/`
46	Beaumont-la-Ronce	990	*
47	Neuville-sur-Brenne	620	*
48	Dolus-le-Sec	540	*
49	Chenonceaux	325	*
50	Candes-Saint-Martin	227	`http://candes-st-martin.fr/`

TAB. B.1 – Echantillon de communes d'Indre-et-Loire.

N° directives	N° des sites web																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	31	30	7	5	4		20	2	2			30	67	1	37		4	7		67
2																				
3	52	54	32	42	27	2	49	12	359	2	467	27	60	6	48	1	49	47	130	66
4	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5		32																		
6											4									
7			1									1					1			
8	2			2	2						2	2	2		2		2	2	2	2
9	2	1	1	2	2	1	1	1	1	1	2	2	2	1	2	1	2	2	2	2
10	6	6		1	5				3		2			6	4					
11	57	55	32	41	26	1	46	12	360	1	454	25	60	4	46		48	46	131	65
12		1	1		1	4		1	1	3	2	1		1	1	3				
13																				
14																				
Total	151	180	74	94	68	9	117	29	727	8	934	89	192	20	141	6	107	105	266	203

TAB. B.2 – Nombre d'erreurs par directive des sites web des communes de plus de 4000 habitants.

N° directives	N° des sites web										
	21	22	24	25	26	28	29	33	34	35	36
1		1		1	18	5	1	6		61	4
2											
3	3	2	1	2	8	58	19	76	5	138	101
4	1	1	1	1	1	1	1	1	1	1	1
5								5			
6		1		1						3	
7								1			
8					2					2	
9	1	1	1	1	2	1	1	1	1	2	1
10					2			1		2	4
11	2	1		1	5	56	17	72	7	139	100
12	4	4	3	4	1	1					1
13											
14											
Total	11	11	6	11	39	122	39	163	14	348	212

TAB. B.3 – Nombre d'erreurs par directive des sites web des communes d'entre 2000 et 4000 habitants.

N° directives	N° des sites web			
	39	40	45	50
1			27	
2				
3	18	2	21	1
4	1	1	1	1
5				
6			2	
7				
8			2	
9	1	1	2	1
10				
11	22	1	22	
12	1	3	1	3
13				
14				
Total	43	8	78	6

Tab. B.4 – Nombre d'erreurs par directive des sites web des communes de moins de 2000 habitants.

N° directives	N° des sites web																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	4	1	0	0	3	1	5	1	97	1	28	0	6	5	5	1	9	0	10	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	6	6	6	5	5	4	6	5	5	4	5	5	18	6	6	4	6	6	6	5
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2
5	22	42	7	7	0	0	16	7	181	0	130	7	4	4	25	0	31	25	55	7
6	13	11	7	4	10	2	5	7	19	2	20	4	23	5	7	3	17	5	11	7
7	20	14	6	5	14	0	8	11	32	0	29	4	32	8	11	0	18	5	14	8
8	2	0	0	0	1	0	2	0	4	0	0	0	0	2	2	0	2	0	0	0
9	3	1	1	1	2	1	3	1	5	1	1	1	1	3	3	1	3	1	1	1
10	2	3	3	3	5	0	2	2	16	3	7	2	61	2	3	0	3	2	4	11
11	2	3	2	2	3	2	2	2	2	2	3	2	3	2	2	3	2	2	2	2
12	1	1	0	1	0	1	1	0	1	2	0	0	1	0	0	1	1	1	1	1
13	10	13	10	10	11	9	11	9	10	9	10	9	10	10	10	10	10	10	10	10
14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Total	92	102	49	45	61	27	68	52	379	31	240	41	166	54	81	30	109	64	122	60

Tab. B.5 – Nombre de warnings par directive des sites web des communes de plus de 4000 habitants.

N° directives	N° des sites web										
	21	22	24	25	26	28	29	33	34	35	36
1	1	0	1	0	0	6	5	0	5	1	2
2	2	2	2	2	2	2	2	2	2	2	2
3	4	4	4	4	6	6	5	6	6	6	7
4	2	2	2	2	2	2	2	2	2	2	2
5	0	0	0	0	10	10	4	16	4	43	16
6	2	1	3	1	2	7	4	5	11	6	4
7	0	0	0	0	2	11	8	5	14	8	5
8	0	0	0	0	0	2	2	0	0	0	1
9	1	1	1	1	1	3	3	1	1	1	2
10	0	0	0	0	1	3	2	2	2	6	2
11	2	2	3	2	3	2	2	3	3	2	2
12	2	0	1	0	1	0	1	1	1	1	1
13	9	9	10	9	10	10	10	10	10	12	10
14	3	3	3	3	3	3	3	3	3	3	3
Total	28	24	30	24	43	67	53	56	64	93	59

Tab. B.6 – Nombre de warnings par directive des sites web des communes d'entre 2000 et 4000 habitants.

N° directives	N° des sites web			
	39	40	45	50
1	15	1	1	1
2	2	2	2	2
3	6	4	6	4
4	2	2	2	2
5	7	0	16	0
6	7	2	8	3
7	11	0	11	0
8	1	0	0	0
9	2	1	1	1
10	2	0	2	0
11	2	2	2	3
12	0	1	0	1
13	9	9	10	10
14	3	3	3	3
Total	69	27	64	30

Tab. B.7 – Nombre de warnings par directive des sites web des communes de moins de 2000 habitants.

N° directives	Nombre d'erreurs	Nombre de warnings
1	438	217
2	0	68
3	1987	197
4	34	71
5	37	696
6	11	248
7	4	314
8	28	21
9	49	56
10	42	156
11	1955	80
12	46	25
13	0	348
14	0	105

Tab. B.8 – Nombre d'erreurs et de warnings par directive.

Annexe C

Evaluation de l'accessibilité des sites web publics d'Indre-et-Loire après utilisation du correcteur automatique

Afin de tester les performances du correcteur des normes d'accessibilité, nous avons réévalué l'accessibilité des sites web présentés dans l'annexe B, après correction des spécifications 1.1, 5.3 et 5.5 des WCAG.

Les tableaux C.1 à C.6 représentent les résultats de chaque site web évalué par notre validateur après avoir passé le correcteur automatique (section 6.2 page 148). Le nombre d'erreurs ou de warnings est indiqué par directive WCAG et le tableau C.7 dresse le nombre d'erreurs et de warnings détectés par directive pour l'ensemble des sites web.

N° directives	N° des sites web																	
	1	2	3	4	5	6	7	8	10	11	12	13	15	16	17	18	19	20
1	0	0	0	0	0	0	1	1	0	0	12	0	2	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	47	2	21	19	27	3	45	7	2	471	7	53	26	1	26	28	25	14
4	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
8	2	0	0	2	2	0	0	0	0	2	2	2	2	0	2	2	2	2
9	2	1	1	2	2	1	1	1	1	2	2	2	2	1	2	2	2	2
10	6	0	0	1	5	0	0	0	0	2	0	0	4	0	0	0	1	0
11	52	1	21	19	26	1	42	8	1	447	7	54	25	0	25	28	26	14
12	0	1	1	0	1	1	0	1	3	2	1	0	1	3	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	110	6	45	44	64	7	90	19	8	932	33	112	63	6	58	61	57	33

TAB. C.1 – Nombre d'erreurs, après correction, par directive des sites web des communes de plus de 4000 habitants.

N° directives	N° des sites web									
	21	22	24	25	28	29	33	34	35	36
1	0	1	0	1	1	1	0	0	1	1
2	0	0	0	0	0	0	0	0	0	0
3	3	2	1	2	18	14	51	5	110	87
4	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0
6	0	1	0	1	0	0	0	0	3	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	0	2	0
9	1	1	1	1	1	1	1	1	2	1
10	0	0	0	0	0	0	1	0	2	4
11	2	1	0	1	17	13	48	7	111	88
12	4	4	3	4	1	0	0	0	0	1
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
Total	11	11	6	11	39	30	103	14	232	183

TAB. C.2 – Nombre d'erreurs, après correction, par directive des sites web des communes d'entre 2000 et 4000 habitants.

N° directives	N° des sites web			
	39	40	45	50
1	0	0	0	0
2	0	0	0	0
3	18	2	21	1
4	1	1	1	1
5	0	0	0	0
6	0	0	2	0
7	0	0	0	0
8	0	0	2	0
9	1	1	2	1
10	0	0	0	0
11	22	1	22	0
12	1	3	1	3
13	0	0	0	0
14	0	0	0	0
Total	43	8	51	6

TAB. C.3 – Nombre d'erreurs, après correction, par directive des sites web des communes de moins de 2000 habitants.

N° directives	N° des sites web																
	1	2	3	4	5	6	7	8	10	11	12	13	15	16	17	18	19	20
1	35	0	9	12	19	1	24	2	1	38	20	73	43	1	40	52	10	68
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	5	4	5	5	5	6	5	5	4	5	6	17	5	4	6	6	6	5
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2
5	0	0	0	0	0	0	0	0	0	37	4	0	0	0	5	16	4	0
6	11	2	7	5	10	2	5	8	2	22	4	23	7	3	17	5	11	7
7	17	0	6	5	14	2	8	11	0	32	4	32	11	0	18	5	14	8
8	2	0	0	0	1	0	2	0	0	0	0	0	2	0	2	0	0	0
9	3	1	1	1	2	1	3	1	1	1	1	1	3	1	3	1	1	1
10	1	0	2	2	5	0	1	1	3	7	2	62	2	0	3	2	6	10
11	2	3	2	2	3	3	2	2	2	3	2	3	2	3	2	2	2	2
12	1	0	0	1	0	0	1	0	2	0	0	1	0	1	1	1	1	1
13	10	10	10	10	11	10	11	9	9	10	9	10	10	10	10	10	10	10
14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Total	94	27	49	50	77	32	69	46	31	162	59	229	92	30	114	107	73	119

TAB. C.4 – Nombre de warnings, après correction, par directive des sites web des communes de plus de 4000 habitants.

N° directives	N° des sites web									
	21	22	24	25	28	29	33	34	35	36
1	1	0	1	0	9	5	10	5	79	5
2	2	2	2	2	2	2	2	2	2	2
3	4	4	4	4	5	5	5	5	6	8
4	2	2	2	2	2	2	2	2	2	2
5	0	0	0	0	0	0	0	0	10	4
6	2	1	3	1	7	5	5	11	6	4
7	0	0	0	0	11	8	5	14	8	5
8	0	0	0	0	2	2	0	0	0	1
9	1	1	1	1	3	3	1	1	1	2
10	0	0	0	0	1	1	1	1	6	2
11	2	2	3	2	2	2	3	3	2	2
12	2	0	1	0	0	1	1	1	1	1
13	9	9	10	9	10	10	10	10	12	10
14	3	3	3	3	3	3	3	3	3	3
Total	28	24	30	24	57	49	48	58	138	51

TAB. C.5 – Nombre de warnings, après correction, par directive des sites web des communes d'entre 2000 et 4000 habitants.

N° directives	N° des sites web			
	39	40	45	50
1	15	1	29	1
2	2	2	2	2
3	5	4	5	4
4	2	2	2	2
5	0	0	0	0
6	7	2	8	3
7	11	0	11	0
8	1	0	0	0
9	2	1	1	1
10	1	0	1	0
11	2	2	2	3
12	0	1	0	1
13	9	9	10	10
14	3	3	3	3
Total	60	27	74	30

TAB. C.6 – Nombre de warnings, après correction, par directive des sites web des communes de moins de 2000 habitants.

N° directives	Nombre d'erreurs	Nombre de warning
1	23	609
2	0	62
3	1159	172
4	31	65
5	0	80
6	12	216
7	4	260
8	26	15
9	45	47
10	26	123
11	1130	74
12	40	21
13	0	316
14	0	96

TAB. C.7 – Nombre d'erreurs et de warnings par directive après correction.

Références bibliographiques

[1] Guide barème pour l'évaluation des déficiences et incapacités des personnes handicapées. Décrets n° 931216 et 931217 du 4 novembre 1993 et le décret n°771549 du 31 décembre 1977.

[2] Loi n° 75-534 du 30 juin 1975 d'orientation en faveur des personnes handicapées. Journal Officiel de la République Française (1er juillet 1975), 1975.

[3] Loi n° 2005-102 du 11 février 2005 pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées. JO n° 36 du 12 février 2005 page 2353, 2005.

[4] A. Afakir. Optimisation de clavier virtuel pour personnes handicapées par des fourmis artificielles. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2008.

[5] A. Amberg, W. Domschke, and S. VoB. Capacitated Minimum Spanning Trees : Algorithms using intelligent search. *Combinatorial Optimization : Theory and Practice*, 1(1) :9–39, 1996.

[6] Assemblée générale de l'ONU. Déclaration des droits des personnes handicapées. proclamée le 9 décembre 1975, 1975.

[7] Assemblée Nationale. Projet de loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées. In *Texte adopté n°377*, 2005.

[8] Y. Bau, C. Ho, and H. Ewe. An Ant Colony Optimization Approach to the Degree-Constrained Minimum Spanning Tree problem. In *Lecture Notes in Computer Science*, volume 3801, pages 657–662, 2005.

[9] G. Beugnon, S. Chameron, N. Monmarché, and G. Venturini. Systèmes de navigation autonomes : de la fourmi au robot. In C. Thinus-Blanc and J. Bullier, editors, *Agir dans l'espace*, Cognitique, chapter 10, pages 193–213. édition de la maison des sciences de l'homme edition, 2005. ISBN 2-7351-1087-7.

[10] A. Blais. Conception d'un module de restructuration de pages web afin qu'elles respectent les normes d'accessibilité. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2007.

[11] M. E. Bobillier-Chaumon and F. Sandoz-Guermond. Apports croisés des démarches d'inspection et de test d'usage dans l'évaluation de l'accessibilité de e-services. In *Actes de congrès ERGO IA'2006*, pages 11–13, Biarritz, octobre 2006.

[12] P. Boissière and D. Dours. Vitipi : Un système d'aide à l'écriture basé sur un principe d'autoapprentissage et adapté à tous les handicaps moteurs. In *IFRATH Handicap 2000*, pages 81–96. AACCESS Reprographie, Paris 15 - 16 Juin 2000.

[13] P. Boissière and D. Dours. Comment VITIPI un système d'assistance à l'écriture pour les personnes handicapées peut offrir des propriétés intéressantes pour le TALN ? In *TALN*, pages 183–192, Tours, 2-5 juillet 2001.

[14] J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems, 1995.

[15] B. Brunk. Overview and Preview Tools for Navigating the World-Wide Web, 1999. SILS Technical Report TR-1999-03.

[16] B. Bullnheimer, R. Hartl, and C. Strauss. Applying the Ant System to the Vehicle Routing Problem. In *Proceedings of the second Metaheuristics International Conference (MIC'97)*, Sophia-Antipolis, France, 1997.

[17] B. Bullnheimer, R. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. In *Sixth Viennese workshop on Optimal Control, Dynamic Games, Nonlinear Dynamics and Adaptive Systems*, Vienna (Austria), Mai 1997.

[18] J. Bulucua. Réorganisation automatique de pages web pour les rendre accessibles. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2008.

[19] L. Caccetta, B. Lam, and S. Hill. Heuristics for the Degree restricted Minimum Spanning Tree problem, 2000.

[20] B. Caldwell, M. Cooper, L. Guarino Reid, and G. Vanderheiden. Techniques for Web Content Accessibility Guidelines 2.0. Working Draft, avril 2008.

[21] E. Cambois, A. Désesquelles, and J.-F. Ravaud. Femmes et hommes ne sont pas égaux face au handicap. In *Population et sociétés, n°386*, janvier 2003.

[22] Centre Technique National d'Etudes et de Recherches sur les Handicaps et les Inadaptations (CTNERHI), la Direction Générale de l'Action Sociale (DGAS), and la Direction de la Recherche, des Etudes, de l'Evaluation et des Statistiques (DREES). Le handicap en chiffres. février 2004.

[23] P. Chalasani and R. Motwani. Approximating capacitated routing and delivery problems. *SIAM Journal on Computing*, 28(6) :2133–2149, 1999.

[24] F. Chapireau, F. Casadebaig, A. Philippe, and D. Ruffin. Description des populations des institutions psychiatriques dans l'enquête HID - Rapport final. In *Document de travail, série Etudes n°44, DREES*, Juillet 2004.

[25] M. Charikar, S. Khuller, and B. Raghavachari. Algorithms for capacitated vehicle routing. pages 349–358, 1998.

[26] A. Chevalier, M. Kicka, and J. Cegarra. Quels sont les effets de la qualité ergonomique d'un site web et de l'expérience des utilisateurs sur la charge cognitive et le temps de navigation ? In *JETCSIC'2004 10e Journée d'Étude sur le Traitement Cognitif des Systèmes d'Information Complexes, Genève*, Juin 2004.

[27] W. Chisholm, G. Vanderheiden, and I. Jacobs. *Web Content Accessibility Guideline 1.0, WCAG 1.0*, 1999. W3C Recommendation 5-May-1999.

[28] W. Chisholm, G. Vanderheiden, and I. Jacobs. CSS Techniques for Web Content Accessibility Guidelines 1.0, Novembre 2000.

[29] W. Chisholm, G. Vanderheiden, and I. Jacobs. HTML Techniques for Web Content Accessibility Guidelines 1.0, Novembre 2000.

[30] W. Chisholm, G. Vanderheiden, and I. Jacobs. Techniques for Web Content Accessibility Guidelines 1.0, décembre 2000.

[31] Christofides, Mingozzi, and Toth. Branch-and-bound (k-shortest spanning trees, q-paths). Mathematical Programming, 1981.

[32] S. Colas, J. Bulucua, N. Monmarché, and M. Slimane. Accessibilité des informations pertinentes des sites web accrue pour les personnes déficientes visuelles par extraction d'informations. In *11ème Colloque Internationale sur le Document Electronique (CIDE 11)*, Rouen (France), 28-31 octobre 2008.

[33] S. Colas, N. Monmarché, D. Burger, and S. Mohamed. A web site migration support tool to reach european accessibility standards. In *9th European conference for the Advancement of Assistive Technology in Europe (AAATE'07), Assistive Technology Research Series*, volume 20, pages 907–911, San Sebastian (Spain), October 2007.

[34] S. Colas, N. Monmarché, P. Gaucher, and M. Slimane. Optimisation de l'organisation d'un clavier virtuel pour personnes handicapées par des fourmis artificielles. In *Métaheuristiques (META'06)*, Hammamet (Tunisie), 2-4 novembre 2006.

[35] S. Colas, N. Monmarché, P. Gaucher, and M. Slimane. Accessibility of french publics web sites : a new tool and methodology to reach accessibility compliance. In *International Conference on Human-Machine iNteraction (HuMaN'07)*, pages 55–60, Timimoun (Algérie), 12-14 mars 2007.

[36] S. Colas, N. Monmarché, P. Gaucher, and M. Slimane. Artificial ants for the optimization of virtual keyboard arrangement for disable people. In *Lecture Notes in Computer Science 4926, Springer, Eds. N. Monmarché, E. Talbi, P. Collet, M. Schoenauer, E. Lutton, 8th International Conference on Artificial Evolution (EA'07)*, pages 87–99, Tours (France), October 2008.

[37] S. Colas, N. Monmarché, and M. Slimane. Accessibilité des sites web institutionnels : mode d'emploi du site d'aide à la migration, application au site de chambray-lès-tours. Rapport interne 291, Laboratoire d'Informatique de l'Université de Tours, juin 2006. 44 pages.

[38] S. Colas, N. Monmarché, and M. Slimane. Génération de plan de site web pour les non-voyants par des fourmis artificielles. In *Métaheuristiques (META'06)*, Hammamet (Tunisie), 2-4 novembre 2006.

[39] S. Colas, N. Monmarché, and M. Slimane. Web browser adapted visually impaired people needs thanks to a shortened braille tool. In *10th Annual Meeting on Health, Science & Technology*, pages 5–8, Université de Tours, 23-24 mai 2006.

[40] S. Colas, N. Monmarché, and M. Slimane. Automatic corrector or webmaster help to reach web accessibility compliance. In *Special issue of the AMSE journals (Association for the Advancement of Modelling and Simulation Techniques in Enterprises)*, 2008.

[41] S. Colas, N. Monmarché, and M. Slimane. Correcteur automatique ou assistance au webmestre pour obtenir un web accessible. In Cépaduès, editor, *Handicap 2008, Eds. Nadine Vigouroux et Philippe Gorce, Cépaduès éditions, ISBN 978-2-85428-848-3*, pages 122–127, France (Paris), 10-12 juin 2008.

[42] S. Colas, N. Monmarché, and M. Slimane. Génération de plan de site web pour les nonvoyants par des fourmis artificielles. In *Revue d'Intelligence Artificielle , Numéro Spécial Métaheuristiques (RIA)*, volume 22, pages 137–159. Hermes, ISBN 978-2-7462-2156-7, 2008.

[43] S. Colas, N. Monmarché, and M. Slimane. Les fourmis artificielles facilitent la navigation sur le web des non-voyants. 18ème Journées Evolutionnaires Trimestrielles (JET18), Paris, mars 2008.

[44] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In *Proceedings of the first European Conference on Artificial Life (ECAL'91), edited by F. Verela and P. Bourgine*, pages 134–142, Cambridge, Mass, USA, MIT Press, 1991.

[45] A. Colvez, H. Gardent, and le Groupe de recherche sur le besoin et les indicateurs en gérontologie. Les indicateurs d'incapacité fonctionnelle en gérontologie : Information, validation, utilisation. Co-édit INSERM/CTNERHI, Paris 1990. 110 pages.

[46] Commission Evolution du Braille Français. Notation mathématique braille - mise à jour de la notation mathématique en braille de 1971 (jusqu'au niveau baccalauréat inclus). Septembre 2001.

[47] R. Dachselt and J. Ebert. Collapsible Cylindrical Trees : A fast hierarchical navigation technique. In *IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 79–86, San Diego, California (USA), 2001.

[48] G. B. Dantzig and R. Ramser. The truck dispatching problem. In *Management Science 6*, pages 80–91, 1959.

[49] M. Dorigo and L. Gambardella. Ant colony sytem : A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1) :53–66, 1997.

[50] M. Dorigo, V. Maniezzo, and A. Colorni. Positive feedback as a search strategy. Technical Report 91-016, Politecnico di Milano, Italy, 1991. 20 pages.

[51] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System : Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1) :29–41, 1996.

[52] M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT Press, Cambridge, MA, USA, 2004.

[53] S. Douglas and A. Happ. Evaluating performance, discomfort, and subjective preference between computer keyboard designs. Salvendy, G. and Smith, M.J. (eds)

Human-Computer Interaction : Software and Hardware Interface, Elsevier, New York, pp 1064-1069, 1993.

[54] M. Dror, M. Haouari, and J. Chaouachi. Generalized spanning trees. In *European Journal of Operational Research*, volume 120, pages 583–592, 2000.

[55] e-mediacité. Etude portant sur l'accessibilité des sites Internet officiels. Audit réalisé du 30 janvier au 6 février 2006 sur un échantillon représentatif de 73 sites publics officiels, 2006.

[56] J. Eggers, D. Feillet, S. Kehl, M.-O. Wagner, and B. Yannou. Optimization of the keyboard arrangement problem using an Ant Colony algorithm. *European Journal of Operational Research*, 148(3) :672–686, August 2003.

[57] T. Ferrand. Coordination motrice inter-individuelle et loi de fitts : Fonction d'échange vitesse-précision pour le mouvement de pointage dyade. In *Cognito*, pages 25–32, 1997.

[58] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology*, volume 47, pages 381–391, 1954.

[59] C. Fontaine. Développement et innovation du braille abrégé. 4ème Symposium scientifique sur l'incapacité visuelle et la réadaptation, février 2001. pp 14-19.

[60] J. Gallant and H. Bergevin. Rendering mode and doctype switching. In *Community MX*, 2004.

[61] M. Galvan. Création d'un outil de migration et de création de sites web accessibles. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2005.

[62] L. Gambardella, E. Taillard, and G. Agazzi. MACS-VRPTW : A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimisation*, pages 63–76. McGraw-Hill, London, UK, 1999.

[63] M. R. Garey and D. S. Johnson. Computers and intractability : A guide to the theory of NP-completeness. Freeman, San Francisco, California, 1979.

[64] M. Gerard, S. Jones, L. Smith, R. Thomas, and T. Wang. An ergonomic evaluation of the kinesis ergonomic computer keyboard. Ergonomics 37, pp 1661-1668, 1994.

[65] J. Germon. Accessibilité du site web du conseil général d'Indre-et-Loire. Rapport de stage, Département d'Informatique (DI) de Polytech'Tours, France, 2005.

[66] F. Glover. Future paths of integer programming and links to artificial intelligence. In *Computers and Operations research*, volume 5, pages 533–549, 1986.

[67] I. Herman, G. Melançon, and M. S. Marshall. Graph Visualization and Navigation in Information Visualization : a Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :24–43, /2000.

[68] B. Hölldobler and E. Wilson. Voyage chez les fourmis. Sciences Ouvertes, Seuil, novembre 1996.

[69] ISO - Organisation Internationale de Normalisation. Iso9241-11 :1998 - Exigences ergonomiques pour travail de bureau avec terminaux à écran de visualisation (TEV) - Partie 11 : Lignes directrices relatives à l'utilisabilité, 1998.

[70] S. Kirkpatrick, C. Gerlatt, and M. Vecchi. Optimization by simulated annealing. In *Science*, volume 220, pages 671–680, 1983.

[71] E. Kommer. Le braille en noir, méthode d'apprentissage du braille intégral destinée aux personnes qui voient. 1994.

[72] E. Kommer. Je donne ma langue au chat, méthode d'abrégé en braille et en noir. 2004.

[73] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society 7*, pages 48–50, 1956.

[74] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1.

[75] D. Langolff, N. Jessel, and D. Levy. MFB (Music For the Blind) : a braille musical software able to transcribe and create musical scores into braille and to be used by blind persons. In *6th ERCIM Workshop "User interfaces for all" , Florence (ITALIE), 25/10/00-26/10/00*, pages 189–194. Pier Luigi Emiliani and Constantine Stephanidis, octobre 2000.

[76] Laporte, Nobert, and Desrochers. Branch-and-cut. Operations Research, 1985.

[77] E. Launay. Création d'un outil de migration et de création de sites web accessibles. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2005.

[78] J. Lazar, A. Dudley-Sponaugle, and K. D. Greenidge. Improving web accessibility : a study of webmaster perceptions. In *Computers in Human Behavior*, volume 20, pages 269–288, 2004.

[79] G. W. Lesher and B. J. Moulton. A method for optimizing single-finger keyboards. In *Proceedings of the RESNA 2000 Annual Conference*, pages 91–93, 2000.

[80] C. Leulier, J. M. C. Bastien, and D. L. Scapin. Compilation of ergonomic guidelines for the design and evaluation of Web sites. In *Commerce & Interaction Report*, Rocquencourt (France) : Institut National de Recherche en Informatique et en Automatique, 1998.

[81] S. Lin. Computer solutions of the travelling salesman problem. In *Bell Systems Technical Journal*, volume 44, pages 2245–2269, 1965.

[82] I. S. MacKenzie and S. X. Zhang. The design and evaluation of a high performance soft keyboard. In ACM, editor, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, pages 25–31, New York, 1999.

[83] I. S. MacKenzie, S. X. Zhang, and R. W. Soukoreff. Text entry using soft keyboards. In *Behaviour & Information Technology*, volume 18, pages 235–244, 1999.

[84] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall : detail and context smoothly integrated. In *CHI '91 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 173–176, New York, NY, USA, 1991. ACM Press. ISBN 0-89791-383-3.

[85] C. Mariage, J. Vanderdonckt, A. Bereikdar, and M. Noirhomme. Destine : outil d'aide à l'évaluation contextualisée de l'ergonomie des sites web.

[86] C. Marin Lamellet, M. P. Bruyas, and L. Guyot. L'utilisabilité d'internet comme source d'information pour les voyageurs handicapés. In *Recherche Transports Sécurité*, number 68, 2000.

[87] C. Marincu and B. McMullin. A comparative assessment of WebAccessibility and technical standards conformance in four EU states. eAccessibility Lab RINCE, Dublin City University, Avril 2004. Final version published in First Monday, volume 9, number 7 (July 2004).

[88] D. Maurel, B. Fouche, and S. Briffault. HandiAS : Aider la communication en facilitant la saisie rapide de textes. In *Handicap 2000 - Nouvelles Technologies : assistance technique aux handicaps moteurs et sensoriel*, pages 87–92, Paris, 15 - 16 Juin 2000.

[89] D. Maurel, N. Rossi, and R. Thibault. HandiAS : Un système multilingue pour l'aide à la communication des personnes handicapées. In *Atelier Thématique TALN 2001*, pages 203–212, Tours, 2-5 juillet 2001.

[90] D. Mboum. Réalisation d'un validateur de l'accessibilité du web. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2007.

[91] C. McKnight, A. Dillon, and J. Richardson. A comparison of linear and hypertext formats in information retrieval. In R. M. A. . C. G. (Eds.), editor, *Hypertext : The state of art*, pages 10–19, Oxford, England, 1990. Intellect Book.

[92] G. Michel, G. Uzan, and G. Voirgard. E-mail tools for all : designing a vocal e-mail to be used by blind people for their electronic messages. In *International Conference on Human-Machine iNteraction (HuMaN'07)*, pages 71–76, Timimoun (Algérie), 12-14 mars 2007.

[93] N. Monmarché, M. Slimane, and G. Venturini. L'algorithme AntClass : classification non supervisée par une colonie de fourmis artificielles. *Extraction des Connaissances et Apprentissage : Apprentissage et évolution*, 1(3) :131–166, 2001.

[94] P. Mormiche. L'enquête HID de l'INSEE : Objectifs et schéma organisationnel. In *Courrier des statistiques*, number 87-88, pages 7–18, décembre 1998.

[95] P. Mormiche and le groupe de projet HID. Le handicap se conjugue au pluriel. In *Insee première*, number 742, octobre 2000.

[96] P. Mormiche and le groupe de projet HID. Le handicap en institution : le devenir des pensionnaires entre 1998 et 2000. septembre 2001.

[97] S. Narula and C. Ho. Degree-constrained minimum spanning trees. In *Computers and Operations Research*, volume 7, pages 239–249, 1980.

[98] D. Nation, C. Plaisant, G. Marchionini, and A. Komlodi. Visualizing Web Sites using a Hierarchical Table of Contents Browser : WebToc. In *Proceedings of Designing for*

the Web : Practices and Reflections (3rd Conference on Human factors and the Web), Denver, 1997.

[99] OMS. International Statistical Classification of Diseases and Related Health Problems, 10th revision (ICD-10), 1993.

[100] OMS. CIH-2 : Classification internationale du fonctionnement et de l'incapacité. Genève, 1999. version provisoire Bêta-2, version complète.

[101] T. Oogane and C. Asakawa. An interactive method for accessing tables in html. In *Proceedings of the third international ACM conference on Assistive technologies (ASSETS'98)*, pages 126–128, New York, 1998.

[102] B. Oriola, F. Amadieu, P. Boissière, M. Raynal, A. Tricot, F. Vella, and N. Vigouroux. Projet ChatCom : Etude des usages de différents claviers logiciels dans une tâche de saisie via internet par des personnes handicapées. In Cépaduès, editor, *Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps (ASSISTH 2007)*, pages 233–240, Toulouse, France, 19-21 novembre 2007.

[103] P. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem : a survey and recent developments. In *DIMACS Series, Discrete Mathematics and Theoretical Computer Science*, volume 16, pages 1–42, 1994.

[104] R. Patterson, H. Pirkul, and E. Rolland. A memory adaptive reasoning technique for solving the Capacitated Minimum Spanning Tree problem. In *Journal of Heuristics 5*, pages 159–180, 1999.

[105] F. Poirier, J.-Y. Antoine, and Z. Barhoumi. Sibylle : améliorations de l'interface à partir de critères ergonomiques et des besoins des utilisateurs. In Cépaduès, editor, *Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps (ASSISTH 2007)*, pages 241 – 247, Toulouse, France, 19-21 novembre 2007.

[106] R. Prim. Shortest connection networks and some generalizations. In *Bell Systems Technology Journal 36*, pages 1389–1401, 1957.

[107] D. Ragget, A. Le Hors, and I. Jacobs. *HTML 4.01 Specification.* W3C Recommendation 24 December 1999.

[108] M. Raynal. Le système KeyGlass : aide à la saisie de caractères pour personnes handicapées moteur. retours sur une première expérimentation. In *Handicap'2006*, Paris, 7-9 juin 2006.

[109] M. Raynal and N. Vigouroux. KeyGlasses : Semi-transparent keys to optimize text input on virtual keyboard. In A. Pruski and H. Knops, editors, *Assistive technology : from virtuality to reality - 8th European conference for the advancement of assistive technology in europe (AAATE 2005) , Lille, France, 06/09/05-09/09/05*, pages 713–717, ISBN 1-58603-543-6, septembre 2005. IOS Press.

[110] M. Reimann, K. Doerner, and R. Hartl. Insertion based Ants for Vehicle Routing Problems with Backhauls and Time Window. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Proceedings of the Third International Workshop on Ant Algorithms*

(ANTS'2002), volume 2463 of *Lecture Notes in Computer Science*, pages 135–148, Brussels, Belgium, September 12-14 2002. Springer Verlag.

[111] M. Reimann, K. Doerner, and R. Hartl. D-Ants : Savings Based Ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4) :563–591, April 2004.

[112] M. Reimann and M. Laumanns. A hybrid aco algorithm for the capacitated minimum spanning tree problem. In *First Internationnal Workshop on Hybrid Metaheuristics (HM 2004)*, 2004.

[113] M. Reimann, M. Stummer, and K. Doerner. A savings based Ant System for the Vehicle Routing Problem. In *Proceedings of GECCO2002*, pages 1317–1325, 2002.

[114] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone Trees : animated 3d visualizations of hierarchical information. In *CHI '91 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 189–194, New York, NY, USA, 1991. ACM Press. ISBN 0-89791-383-3.

[115] G. Salton and M. McGill. Introduction to modern information retrieval. In *McGraw-Hill Book Company*, 1983.

[116] F. Sandoz-Guermond and M.-E. Bobillier-Chaumon. L'accessibilité des e-services aux personnes non-voyantes : difficultés d'usage et recommandations. In *Actes du Congrès international IHM'2006*, pages 35–39, Montréal, avril 2006.

[117] F. Sandoz-Guermond and M.-E. Bobillier-Chaumon. L'accessibilité des nouvelles technologies (e-services) : un enjeu pour l'intégration sociale des personnes handicapées. In *Actes du Congrès international de la SELF'2006*, Caen, septembre 2006.

[118] D. L. Scapin and J. M. C. Bastien. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. In *Behaviour & Information Technology*, volume 17, pages 220 – 231, 1997.

[119] I. Schadle, J.-Y. Antoine, B. Le Pévédic, and F. Poirier. SibyLettre : prédiction de lettre pour la communication assistée. In *Revue d'Interaction Homme-Machine, RIHM*, volume 3, 2002.

[120] I. Schadle, B. Le Pévédic, J.-Y. Antoine, and F. Poirier. SibyLettre : Système de prédiction de lettre pour l'aide à la saisie de texte. In *Atelier Thématique TALN 2001*, pages 233 – 242, Tours, 2-5 Juillet 2001.

[121] Secrétariat d'Etat chargé de la prospective, de l'évaluation des politiques publiques, et du développement de l'économie numérique. Assises du numérique : 27 pistes de travail ouvertes à la concertation pour préparer le plan de développement de l'économie numérique, mai 2008. Mesure 4 du document.

[122] Y. Sharaiha, M. Gendreau, G. Laporte, and I. Osman. A tabu search algorithm for the capacitated shortest spanning tree problem. In *Networks 29*, pages 161–171, 1997.

[123] S. Shyu, P. Yin, B. Lin, and M. Haouari. Ant-Tree : an ant colony approach to the generalized minimum spanning tree problem. *Journal of Experimental & theorical Artificial Intelligence*, 15 :103–112, 2003.

[124] P. Siarry and G. Dreyfus. La méthode du recuit simulé : théorie et application. In *ESPCI-IDSET, 10 rue Vauquelin, Paris*, 1989.

[125] C. Stephanidis and S. Akoumianakis. Accessibility guidelines and scope of formative HCI design input : contrasting two perspectives. In *5th ERCIM Workshop on User Interfaces for All*, 1999.

[126] T. Stützle and H. Hoos. MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In *IEEE*, pages 308–313, 1997.

[127] T. Sullivan and R. Matson. Barriers to use : Usability and content accessibility on the web's most popular sites. In *Proceedings on the 2000 conference on Universal Usability*, 2000.

[128] N. G. Swanson, T. L. Galinsky, L. L. Cole, C. S. Pan, and S. L. Sauter. The impact of keyboard design on comfort and productivity in a text-entry task. Applied Ergonomics, Volume 28, Issue 1, pp 9-16, February 1997.

[129] B. Thomas. Migration vers l'accessibilité aux malvoyants de sites web institutionnels. Rapport de Master 2 Recherche, Département d'Informatique (DI) de Polytech'Tours, France, 2006.

[130] B. Thomas. Migration vers l'accessibilité aux malvoyants de sites web institutionnels. Rapport de projet de fin d'études, Département d'Informatique (DI) de Polytech'Tours, France, 2007.

[131] G. Uzan and N. Jobard. Difficultés rencontrées par les aveugles et déficients visuels pour la consultation de sites web sur les transports et le tourisme. sous la direction de J.C Sperandio, rapport de recherche LEI/INEREC pour l'Institut pour la Ville en Mouvement, Paris, 2002.

[132] F. Vella and N. Vigouroux. Disposition spatiale des touches/caractères des claviers logiciels et fatigue motrice : premiers résultats expérimentaux. In *Handicap 2006*, pages 167–172, Paris, 6-9 juin 2006.

[133] F. Vella, N. Vigouroux, and P. Truillet. SOKEYTO : a design and simulation environment of software keyboards. In A. Pruski and H. Knops, editors, *Assistive technology : from virtuality to reality - 8th European conference for the advancement of assistive technology in europe (AAATE 2005)*, pages 723–727, Lille, France, 6-9 septembre 2005. IOS Press ISBN 1-58603-543-6.

[134] N. Vigouroux, F. Vella, G. Lepicard, D. Chêne, N. Biard, and A. Massonneau. Utilisabilité des sites web pour les personnes handicapées moteurs. In Cépaduès, editor, *Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps (ASSISTH 2007)*, pages 248 – 253, Toulouse, France, 19-21 novembre 2007.

[135] WAI and Authoring Tool Accessibility Guideline Working Group (ATAG WG). Authoring Tool Accessibility Guideline 1.0, ATAG 1.0. W3C Recommendation 3-Febuary-2000.

[136] WAI and Authoring Tool Accessibility Guideline Working Group (ATAG WG). Authoring Tool Accessibility Guideline 2.0, ATAG 2.0. Working Draft, mars 2008.

[137] WAI and User Agent Accessibility Guideline Working Group (UAWG). User Agent Accessibility Guideline 1.0, UAAG 1.0. W3C Recommendation 17-December-2002.

[138] WAI and User Agent Accessibility Guideline Working Group (UAWG). User Agent Accessibility Guideline 2.0, UAAG 2.0. Working Draft, mars 2008.

[139] WCAG WG. *Web Content Accessibility Guideline 2.0, WCAG 2.0*, 2008. W3C Candidate Recommendation 30 April 2008.

[140] H. Wium Lie and B. Bos. *Cascading Style Sheets, level 1.* W3C Recommendation 17 Dec 1996, revised 11 Jan 1999.

[141] P. Zaphiris and R. Ellis. Website Usability and Content Accessibility of the top USA Universities. 2001.

[142] P. Zaphiris and S. H. Kurniawan. Usability and accessibility comparison of governemental, organizational, educational and commercial aging/health-related web sites. 2001.

[143] P. Zaphiris and H.-Y. Terry Ma. The Usability and Content Accessibility of the e-Government in the UK. 2001.

[144] S. Zhai, M. Hunter, and B. A. Smith. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In ACM., editor, *Proceedings of The 13th Annual ACM Symposium on User Interface Software and Technology (UIST).*, pages 119–218, San Diego, California, 2000.

[145] S. X. Zhang. A high performance soft keyboard for mobile systems. The University of Guelph, 1998.

Références Internet

[146] La disposition de clavier fr-dvorak-bépo. http://www.clavier-dvorak.org/wiki/Accueil - dernier accès le 23/06/2008.

[147] Projet Drempels Weg aux Pays Bas. http://www.drempelsweg.nl - dernier accès le 23/11/2006.

[148] Projet See-it-Right. Disponible sur le site du Royal National Institute for the Blind, http://www.rnib.org.uk/xpedio/groups/public/documents/publicWebsite/public_seeitright.hcsp - dernier accès le 03/07/2008.

[149] Questionnaire des différentes collectes d'informations de l'enquête HID. http://rfr-handicap.inserm.fr/HID/questionnaires_HID.htm - dernier accès le 24/10/2006.

[150] Site RFRH de référence sur l'enquête HID. http://rfr-handicap.inserm.fr/HID/ACCUEIL_HID_NEW.HTM - dernier accès le 24/10/2006.

[151] SitemapDoc - Google Sitemap Generator and Editor. http://www.sitemapdoc.com/ - dernier accès le 04/07/2008.

[152] Winbraille, logiciel de transcription de texte en braille. http://handy.univ-lyon1.fr/projets/bramanet/ressources/Installation%20de%20Winbraille%203.pdf - dernier accès le 23/06/2008.

[153] Loi D.D.A. (Disability Discrimination Act) en Grande-Bretagne, 1995. http://www.opsi.gov.uk/acts/acts1995/Ukpga_19950050_en_1.htm - dernier accès le 25/05/2008.

[154] Loi anti-discrimination du 25 février 2003 en Belgique, 2003. http://www.diversite.be/index.php?action=wetgeving_detail\&id=16\&select_page=12 - dernier accès le 05/07/2008.

[155] Etude MeAC (Measuring Progress of eAccessibility in Europe), octobre 2007. http://ec.europa.eu/information_society/activities/einclusion/library/studies/meac_study/index_en.htm - dernier accès le 06/07/2008.

[156] Loi anti-discrimination du 10 mai 2007 supprimant celle du 25 février 2003 en Belgique, 2007. http://www.diversite.be/index.php?action=wetgeving_detail\&id=15\&select_page=12 - dernier accès le 05/07/2008.

[157] ABF Software. Lens. http://www.abf-soft.com/lens-magnifying-glass.shtml - dernier accès le 07/07/2008.

[158] Accessible Information Solution (AIS). Web accessibility toolbar. Pour Internet Explorer. http://www.wat-c.org/tools/WAT/versions/fr/about.html - dernier accès le 19/06/2008.

[159] AccessKeys. AccessColor. http://www.accesskeys.org/tools/color-contrast.html - dernier accès le 19/06/2008.

[160] Accès pour tous. Guide accessibilité. http://www.acces-pour-tous.net/fichiers_communs/accessibilite.php - dernier accès le 19/12/2006.

[161] ALDEIS. Yooda Map - Logiciel d'édition du fichier Sitemap. http://www.yooda.com/outils_referencement/Yooda_map.php - dernier accès le 04/06/2008.

[162] AnySurfer. Label de qualité attestant de l'accessibilité des sites internet (anciennement blindsurfer). http://www.anysurfer.be/fr/ - dernier accès le 10/06/2008.

[163] AssistiveWare. Keystrokes. http://www.assistiveware.com/keystrokes.php - dernier accès le 30/06/2008.

[164] ATRC University Of Toronto. A-Checker. http://checker.atrc.utoronto.ca/index.html - dernier accès le 19/12/2007.

[165] ATRC University Of Toronto. A-Prompt. http://www.aprompt.ca - dernier accès le 19/05/2008.

[166] BAUM. VIRGO, logiciel lecteur d'écran. http://www.baum.fr - dernier accès le 20/06/2008.

[167] BigKeys Company. Clavier bigkeys. http://www.bigkeys.com/default.asp - dernier accès le 10/05/08.

[168] Bjoern. Another Lens, logiciel d'agrandissement, 1998. http://www.b-zone.de/software/lens.htm - dernier accès le 23/06/2008.

[169] V. Bénard. 11 règles de base pour créer des sites faciles à utiliser, mars 2002. http://www.veblog.com/fr/2000/0816-utilisabilite-regles.html - dernier accès le 10/05/2008.

[170] Braillenet. Barre d'outil Accessiweb. http://www.accessiweb.org/fr/groupe_travail_accessibilite_du_web/barre_accessiweb/ - dernier accès le 19/05/2008.

[171] Braillenet. Guide Accessiweb. http://www.accessiweb.org/fr/guide_accessiweb/index.html - dernier accès le 10/07/2008.

[172] Braillenet. Label Accessiweb contrôlant l'accessibilité des sites web. http://www.accessiweb.org/fr/Label_Accessibilite/ - dernier accès le 10/06/2008.

[173] Braillenet. Schéma de migration vers une accessibilité du web harmonisée en Europe (version française de Braillenet). http://www.accessiweb.org/_repository/files/article_harmonisation_accessibilite_web_040906.pdf - dernier accès le 19/06/2008.

[174] Braillenet. Site de l'association Braillenet. http://www.braillenet.org - dernier accès le 23/06/2008.

[175] Braillenet. Tableau de correspondance Accessiweb/WCAG 1.0. http://www.accessiweb.org/fr/guide_accessiweb/edit_table_awv11_fr_wcag10_en.html - dernier accès le 19/06/2008.

[176] Braillenet. Guide Braillenet à l'usage des webmestres pour une meilleure accessibilité des sites publics aux personnes handicapées, octobre 2002. http://www.braillenet.org/accessibilite/guide/ - dernier accès le 19/05/2008.

[177] Braillenet. Tableau de correspondance Accessiweb/UWEM 1.0, septembre 2006. Correspondance avec Accessiweb 1.0 en 2006, Correspondance avec Accessiweb 1.1 en 2008. http://www.accessiweb.org/fr/guide_accessiweb/edit_table_awv11_fr_uwem10_fr.html - dernier accès le 19/06/2008.

[178] Braillenet. Traduction française de l'UWEM 1.0, septembre 2006. http://www.accessiweb.org/fr/uwem/index.html - dernier accès le 19/05/2008.

[179] Braillenet. Proposition de décret de l'article 47 de la loi n°2005-102 par l'association BrailleNet au 7 décembre 2007, décembre 2007. http://www.accessiweb.org/fr/accessibilite_web/actualites/decret47_BrailleNet_accessibilite_numerique_7dec2007/#decret_7dec2007 - dernier accès le 08/06/2008.

[180] M. Brooks. Introducing the dvorak keyboard, 2000. http://www.mwbrooks.com/dvorak/ -dernier accès le 06/07/2008.

[181] D. Burger. La loi sur l'accessibilité va changer la donne dans les appels d'offres. Journal du Net. 10 janvier 2005. http://solutions.journaldunet.com/0501/050110_3questions_braillenet.shtml - dernier accès le 06/05/2008.

[182] Centre des Archives Nationales du monde du travail. De la plume aux claviers : Un siècle de métiers du secrétariat et du bureau. http://www.archivesnationales.culture.gouv.fr/camt/fr/se/secretaire.html - dernier accès le 06/05/2008.

[183] Direction Générale de la Modernisation de l'Etat (DGME). Référentiel accessibilité des services internet de l'administration française. http://www.modernisation.gouv.fr/ - dernier accès le 01/07/2008.

[184] E. Dumout. Les sites publics ont 3 ans pour être accessibles aux handicapés. ZDNet France., Vendredi 4 février 2005. http://zdnet.fr/actualites/internet/0,39020774,39204632,00.htm - dernier accès le 06/06/2008.

[185] Duxbury. DBTWIN, traducteur braille. http://www.avh.asso.fr/magasin/produits/81040.php - dernier accès le 07/07/2008.

[186] Econonet International. Simply Web 2000. http://www.econointl.com/sw/ - dernier accès le 17/06/2008.

[187] Electrum. Powermapper. http://www.powermapper.com/products/mapper/index.htm - dernier accès le 08/06/2008.

[188] Electrum. Sortsite. http://www.powermapper.com/products/sortsite/index.htm - dernier accès le 20/06/2008.

[189] Etre. Color Blindness Simulator. http://www.etre.com/tools/colourblindsimulator/ - dernier accès le 19/05/2008.

[190] EuraCert. European eAccessibility Certification. http://www.euracert.org - dernier accès le 20/06/2008.

[191] Euro Accessibility Consortium. Site officiel. http://www.euroaccessibility.org/ - dernier accès le 10/11/2006.

[192] Europa. eEurope2005 - 2ème plan d'action de l'union européenne. `http://europa.eu/scadplus/leg/fr/lvb/l24226.htm` - dernier accès le 06/05/2008.

[193] Europa. eEurope2010 - 3ème plan d'action de l'union européenne. `http://ec.europa.eu/information_society/eeurope/i2010/index_en.htm` - dernier accès le 10/05/2008.

[194] Europa. Le portail de l'union européenne. `http://europa.eu/` - dernier accès le 06/05/2008.

[195] Freedom Scientific. JAWS. `http://www.freedomscientific.com/fs_products/software_jaws.asp` - dernier accès le07/07/2008.

[196] Fujitsu Limited. Web Accessibility Inspector. `http://www.fujitsu.com/global/accessibility/assistance/wi/` - dernier accès le 19/06/2008.

[197] Fundacion Sidar. Hera. `http://www.sidar.org/hera/` - dernier accès le 19/06/2008.

[198] G2D. Etude et conception de guides doigts sur mesure. `http://guidedoigts.free.fr/` - dernier accès le 16/05/08.

[199] Google software . Google Sitemap Gen, 2005. `http://goog-sitemapgen.sourceforge.net/` - dernier accès le 04/07/2008.

[200] Handicap International. Clavicom. `http://www.handicap-icom.asso.fr/adaptations/aides_techniques/clavicom.html` - dernier accès le 30/05/2008.

[201] HiSoftware. Cynthia Says. `http://www.cynthiasays.com/` - dernier accès le 19/06/2008.

[202] iCITA (Illinois Center for Instructional Technology Accessibility). IE accessible : barre d'outil pour internet explorer. `http://cita.rehab.uiuc.edu/software/ieaccessible/document/toolbar.html` - dernier accès le 17/01/2007.

[203] INSERM, Equipe INOVA (Interfaces Non Visuelles et Accessibilité). BME, Braille Music Editor. `http://www.snv.jussieu.fr/inova/villette2002/act12.htm` - dernier accès le 23/04/2008.

[204] INSERM, Equipe INOVA (Interfaces Non Visuelles et Accessibilité). BrailleStar, logiciel de transcription de texte en braille. `http://inova.snv.jussieu.fr/helene-html/help/macro/dtb-bm-l04-l15-l10.html` - dernier accès le 23/04/2008.

[205] INSERM, Equipe INOVA (Interfaces Non Visuelles et Accessibilité). BrailleSurf. `http://www.snv.jussieu.fr/inova/bs4/index.htm` - dernier accès le 17/03/2008.

[206] In'Tech INFO. Custom Virtual Keyboard (CVK). `http://www.cvk.fr/` - dernier accès le 30/06/2008.

[207] Inxight. VizServer : outil de visualisation de données. `http://labs.businessobjects.com/vizserver/default.asp` - dernier accès le 15/06/2008.

[208] Ixacta. Ixsite Web Analyzer. `http://www.ixacta.com/index.html` - dernier accès le 04/07/2008.

[209] I. Jacobs. Objectifs du World Wide Web Consortium en 7 points, 1999. `http://www.w3.org/Consortium/Points/` - dernier accès le 06/06/2008.

[210] Juicy Studio. Color Constrast Analyser. `http://juicystudio.com/services/colourcontrast.php` - dernier accès le 19/06/2008.

[211] Kinesis Corporation. Computer ergonomics. `http://www.kinesis-ergo.com/` - dernier accès le 8/07/2008.

[212] La Grange. Plongez dans l'accessibilité. `http://www.la-grange.net/accessibilite/index.html` - dernier accès le 19/07/2008.

[213] Laboratory of HCI for Special Needs - UPV/EHU. EvalAccess. `http://sipt07.si.ehu.es/evalaccess2/index.html` - dernier accès le 19/05/2008.

[214] Lake Software. Click-n-type. `http://www.lakefolks.org/cnt/fr-intro.htm` - dernier accèsle 15/06/2008.

[215] A. Le Diberder. Sabir cyber Clavier. Le Monde Interactif, 14 février 2001. `http://www.algo.be/ergo/lemonde-clavier.html` - dernier accès le 06/05/2008.

[216] F. Leboutte. La disposition Dvorak-fr, mai 2002. `http://www.algo.be/ergo/dvorak-fr.html` - dernier accès le 23/05/2008.

[217] Madentec. ScreenDoors 2000. `http://www.madentec.com/products/screendoors.php` - dernier accès le 30/06/2008.

[218] Maltron Keyboards. The ergonomic keyboard specialists. `http://www.maltron.com/maltron-keyboards.html` - dernier accès le 20/05/2008.

[219] G. Mazzocato. Accessibility Color Wheel. `http://gmazzocato.altervista.org/colorwheel/wheel.php` - dernier accès le 07/07/2008.

[220] B. McCann. GOODFEEL Braille Music Translator. Dancing dots Braille Music Technology, 1997. `http://www.dancingdots.com` - dernier accès le 23/06/2008.

[221] Ministère du travail, des relations sociales, de la famille et de la solidarité. Présentation du plan handicap visuel, juin 2008. Mesure 22 du document. `http://www.travail-solidarite.gouv.fr/IMG/pdf/Dossier_de_presse_2JUIN08.pdf` - dernier accès le 30/06/2008.

[222] Mission Handicap (M.H.) de l'Université Claude Bernard Lyon 1. BraMaNet, 2003. `http://handy.univ-lyon1.fr/MH/bramanet/bramanet.php` - dernier accès le 07/07/2008.

[223] Y. Morin. Validateur HTML/XHTML du W3C en français. `http://yansanmo.no-ip.org/ysm-validator/check.php` - dernier accès le 19/06/2008.

[224] Mozdev. Accessibar. `http://accessibar.mozdev.org/` - dernier accès le 19/04/2008.

[225] Mozdev. MozBraille. `http://mozbraille.mozdev.org/index.html` - dernier accès le 17/05/2008.

[226] NICTA (National Information and Communications Technology Australia). GEOmi : outils de visualisation de données. `http://nicta.com.au/director/research/programs/imagen/research_projects/research%20projects_valacon.cfm` - dernier accès le 15/11/2006.

[227] J. Nielsen. Voodoo usability. Jakob Nielsen's Alterbox, décembre 1999. `http://www.useit.com/alertbox/991212.html` - dernier accès le 04/06/2008.

[228] Nullpointer. WebTracer : générateur de plans de sites web. http://www.nullpointer.co.uk/-/webtracer.htm - dernier accès le 15/06/2008.

[229] Oaklett. Faire une sitemap (plan du site) pour Navibar. http://navibar.oaklett.org - dernier accès le 05/01/2008.

[230] OMS. Site français de l'organisation mondiale de la santé. http://www.who.int/fr/ - dernier accès le 24/06/2008.

[231] OMS. Ampleur et causes des déficiences visuelles, 2002. http://www.who.int/mediacentre/factsheets/fs282/fr/ - dernier accès le 24/06/2008.

[232] OMS. Introduction de la version finale de la CIF. 26 pages, 2002. http://www3.who.int/icf/intros/CIF-Fre-Intro.pdf - dernier accès le 24/06/2008.

[233] J. Porter. Testing the Three-Click Rule, avril 2003. http://www.uie.com/articles/three_click_rule/ - dernier accès le 03/07/2008.

[234] Proze. ProzeValide, validateur W3C Multipages. http://www.validateur.ca/ - dernier accès le 19/03/2008.

[235] D. Ragget. Tidy. http://tidy.sourceforge.net/ - dernier accès le 19/05/2008.

[236] Section 508. Loi sur la réadaptation des personnes handicapées ("Rehabilitation Act") au Etats Unis, 1998. http://www.section508.gov/index.cfm?FuseAction=Content\&ID=14 - dernier acès le 10/06/2008.

[237] M. Siebeneicher. Navibar : Extension de navigation pour firefox. https://addons.mozilla.org/fr/firefox/addon/959 - dernier accès le 04/06/2008.

[238] Sitemaps.org. Création de sitemaps. http://www.sitemaps.org/fr/index.php - dernier accès le 04/07/2008.

[239] G. Slinn. Hermish. http://www.hermish.com/index.cfm - dernier accès le 19/02/2008.

[240] SPIP. Site web du système de gestion de contenu SPIP. http://www.spip.net/ - dernier accès le 23/03/2008.

[241] Support EAM, Supporting the creation of an eAccessibility Mark. Site du Support EAM. http://www.support-eam.org/ - dernier accès le 19/03/2008.

[242] Tash. An ablenet company. http://www.ablenetinc.com/tash/index.html - dernier accès le 10/05/08.

[243] Textware Solutions. Fitaly. http://www.fitaly.com/ - dernier accès le 10/05/2008.

[244] TYPO3. Site web du système de gestion de contenu TYPO3. http://typo3.org/ - dernier accès le 23/06/2008.

[245] UB Access. ART Guide. http://www.ubaccess.com/artguide.html - dernier accès le 19/12/2006.

[246] Union Européenne. Information providers guide. http://ec.europa.eu/ipg/index_en.htm - dernier accès le 10/04/2008.

[247] Urbilog/France-Telecom. Ocawa. http://www.ocawa.com - dernier accès le 19/06/2008.

[248] Validome. Validateur HTML/XHTML. http://www.validome.org/ - dernier accès le 19/06/2008.

[249] Virtual Projects. JoyMouse. http://www.vp-soft.com/software/joymouse.php - dernier accès le 6/02/2007.

[250] Voir+. Guide accessibilité. http://www.voirplus.net/acces/ - dernier accès le 19/12/2006.

[251] W3C. CSS Validator. http://jigsaw.w3.org/css-validator/ - dernier accès le 19/06/2008.

[252] W3C. HTML Validator. http://validator.w3.org/ - dernier accès le 19/06/2008.

[253] W3C. A propos du world wide web consortium. http://www.w3.org/Consortium/ - dernier accès le 06/06/2008.

[254] W3Quebec. Validateur HTML/XHTML. http://w3qc.org/validateur/ - dernier accès le 19/06/2008.

[255] WAB Cluster. Site web du WAB Cluster. http://www.wabcluster.org/ - dernier accès le 19/06/2008.

[256] WAB Cluster. UWEM 1.2, aout 2007. http://www.wabcluster.org/uwem1_2/ - dernier accès le 10/07/2008.

[257] WAB Cluster. UWEM 1.0, 5/07/2006. http://www.wabcluster.org/uwem1/ - dernier accès le 19/06/2008.

[258] WAI. Web Accessibility Initiative. http://www.w3.org/WAI - dernier accès le 06/07/2008.

[259] Watchfire. WebXACT (Bobby). http://webxact.watchfire.com/ - dernier accès le 19/12/2006.

[260] Web Design Group (WDG). Validateur HTML du WDG. http://www.htmlhelp.com/tools/validator/ - dernier accès le 19/06/2008.

[261] WebAIM (Web Accessibility In Mind). The Wave. http://www.wave.webaim.org - dernier accès le 19/06/2008.

[262] WebdevOut. Web browser CSS support. http://www.webdevout.com/browser-support-css - dernier accès le 05/05/2008.

Sonia COLAS

OUTILS D'AMÉLIORATION DE L'ACCESSIBILITÉ DU WEB POUR LES PERSONNES VISUELLEMENT HANDICAPÉES

Résumé :

Pour profiter des nombreux services offerts *via* Internet et « surfer » sur la toile comme quiconque, les personnes handicapées ont recours à des aides techniques. Des normes ont été établies pour assurer la compatibilité des sites web avec les aides techniques, et des lois ont été instaurées pour en imposer le respect. Néanmoins encore trop peu de sites web respectent ces normes et sont de ce fait inaccessibles aux personnes handicapées.

Lors de ce travail, nous avons cherché à améliorer l'accessibilité du web aux personnes handicapées, en contribuant à l'adaptation « du » web (ou adaptation du contenu, c'est-à-dire en développant des outils pour aider les webmestres à rendre leur site accessible au regard des normes) ainsi qu'à l'adaptation « au » web (ou adaptation de l'utilisateur).

Mots clés :

informatique pour la compensation des déficits d'autonomie, accessibilité du web, handicap, malvoyant, non-voyant, fourmis artificielles

Abstract :

Disabled persons use assistive technologies to take advantage of numerous services offered by Internet and to navigate on the World Wide Web as whoever. Standards were established to insure compatibility of Web sites with assistive technologies, and laws were established to impose respect of these standards. However few Web sites respect these standards and as a result a lot of Web sites are inaccessible for disabled persons.

During this work, we tried to improve web accessibility to the disabled persons, by contributing to the web adaptation (or contents adaptation, i.e. by proposing tools to help webmasters to return their accessible site towards the standards) as well as to the adaptation to Web (or user adaptation).

Keywords :

computer science to compensate for autonomy deficiency, web accessibility, people with disabilities, visually handicapped people, blind people, artificial ant